日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:　２００１年　１月３１日

出願番号
Application Number:　特願２００１－０２４５１３

[ST.10/C]:　[ＪＰ２００１－０２４５１３]

出願人
Applicant(s):　パシフィック・デザイン株式会社

２００２年　１月１８日

特許庁長官
Commissioner,
Japan Patent Office　及川耕造

出証番号　出証特２００１－３１１７３８９

【書類名】　特許願
【整理番号】　000283P118
【あて先】　特許庁長官殿
【国際特許分類】　G06F　9/20
【発明者】
　【住所又は居所】　東京都新宿区西新宿６丁目１２番１号　パシフィック・デザイン株式会社内
　【氏名】　下郡　慎太郎
【発明者】
　【住所又は居所】　東京都新宿区西新宿６丁目１２番１号　パシフィック・デザイン株式会社内
　【氏名】　鎌野　昇一
【発明者】
　【住所又は居所】　東京都新宿区西新宿６丁目１２番１号　パシフィック・デザイン株式会社内
　【氏名】　北島　利明
【特許出願人】
　【識別番号】　598149242
　【氏名又は名称】　パシフィック・デザイン株式会社
【代理人】
　【識別番号】　100102934
　【弁理士】
　【氏名又は名称】　今井　彰
【手数料の表示】
　【予納台帳番号】　050728
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1

【物件名】　　要約書　1

【プルーフの要否】　　要

【書類名】　　明細書

【発明の名称】　データ処理システム、データ処理装置およびその制御方法

【特許請求の範囲】

【請求項１】　複数のデータ処理装置を有するデータ処理システムであって、２以上の前記データ処理装置は、専用命令により実行される特定のデータ処理用のデータパス部を備えた少なくとも１つの専用データ処理ユニットと、汎用命令により汎用処理を実行可能な汎用データ処理ユニットと、前記専用命令および汎用命令を備えたプログラムに基づき、前記専用データ処理ユニットおよび汎用データ処理ユニットに命令を発行する命令発行ユニットとを有する第１のタイプのデータ処理装置であり、

少なくとも１つの前記第１のタイプのデータ処理装置の汎用データ処理ユニットは、他の前記第１のタイプのデータ処理装置の前記汎用データ処理ユニットとデータを交換可能な通信手段を備えているデータ処理システム。

【請求項２】　請求項１において、少なくとも１つの前記第１のタイプのデータ処理装置の少なくとも１つの前記専用データ処理ユニットは、第２のタイプの前記データ処理装置とデータを交換する機能を備えているデータ処理システム。

【請求項３】　請求項１において、前記第１のタイプのデータ処理装置は、前記プログラムを記憶するコードメモリと、前記汎用命令によりデータを入力または出力可能なデータメモリとを備えており、

前記通信手段は、前記汎用命令に基づき入力または出力するデータの入力アドレスまたは出力アドレスが予め設定されたアドレスのときに、他の前記第１のタイプのデータ処理装置に含まれた前記データメモリに対してデータを入力または出力するようにデータを交換するデータ処理システム。

【請求項４】　請求項３において、前記通信手段は、前記出力アドレスが前記予め設定されたアドレスのときに、他の前記第１のタイプのデータ処理装置へデータを送信する手段を備えているデータ処理システム。

【請求項５】　請求項３において、前記通信手段は、前記入力アドレスが前記予め設定されたアドレスのときに、他の前記第１のタイプのデータ処理装置か

らデータを受信する手段を備えているデータ処理システム。

【請求項６】　請求項３において、前記通信手段は、他の前記第１のタイプのデータ処理装置からデータを受信すると前記データメモリの所定のアドレスにデータを記憶する手段を備えているデータ処理システム。

【請求項７】　請求項６において、前記通信手段は、前記データを記憶する手段がデータを記憶する前記データメモリの受信専用領域が前記汎用データ処理ユニットにより読み出されているときは、前記データを記憶する手段の処理を延期し、前記データを記憶する手段の処理中は、前記汎用データ処理ユニットが前記受信専用領域からデータを読み出す処理を延期する調停手段を備えているデータ処理システム。

【請求項８】　請求項３において、前記通信手段は、他の前記第１のタイプのデータ処理装置からデータを要求されると前記データメモリの所定のアドレスからデータを提供する手段を備えているデータ処理システム。

【請求項９】　請求項８において、前記通信手段は、前記データを提供する手段がデータを取得する前記データメモリの送信専用領域が前記汎用データ処理ユニットにより書き込まれているときは、前記データを提供する手段の処理を延期し、前記データを提供する手段の処理中は、前記汎用データ処理ユニットが前記送信専用領域にデータを書き込む処理を延期する調停手段を備えているデータ処理システム。

【請求項１０】　請求項１において、複数の前記第１のタイプのデータ処理装置の前記専用データ処理ユニットを含む、単一のデータの流れを処理するデータ処理系統が形成されているデータ処理システム。

【請求項１１】　請求項１において、複数の前記第１のタイプのデータ処理装置の前記専用データ処理ユニットを含む、データの流れを処理する複数のデータ処理系統が形成されているデータ処理システム。

【請求項１２】　専用命令により実行される特定のデータ処理用のデータパス部を備えた少なくとも１つの専用データ処理ユニットと、

汎用命令により汎用処理を実行可能な汎用データ処理ユニットと、

前記専用命令および汎用命令を備えたプログラムに基づき、前記専用データ処

理ユニットおよび汎用データ処理ユニットに命令を発行する命令発行ユニットとを有するデータ処理装置であって、

前記汎用データ処理ユニットは、他の前記データ処理装置の前記汎用データ処理ユニットとデータを交換可能な通信手段を備えているデータ処理装置。

【請求項１３】　請求項１２において、前記プログラムを記憶するコードメモリと、前記汎用命令によりデータを入力または出力可能なデータメモリとを有し、

前記通信手段は、前記汎用命令に基づき入力または出力するデータの入力アドレスまたは出力アドレスが予め設定されたアドレスのときに、他の前記データ処理装置との間でデータを交換するデータ処理装置。

【請求項１４】　請求項１３において、前記通信手段は、前記出力アドレスが前記予め設定されたアドレスのときに、他の前記データ処理装置へデータを送信する手段を備えているデータ処理装置。

【請求項１５】　請求項１３において、前記通信手段は、前記入力アドレスが前記予め設定されたアドレスのときに、他の前記データ処理装置からデータを受信する手段を備えているデータ処理装置。

【請求項１６】　請求項１３において、前記通信手段は、他の前記データ処理装置からデータを受信すると前記データメモリの所定のアドレスにデータを記憶する手段を備えているデータ処理装置。

【請求項１７】　請求項１６において、前記通信手段は、前記データを記憶する手段がデータを記憶する前記データメモリの受信専用領域が前記汎用データ処理ユニットにより読み出されているときは、前記データを記憶する手段の処理を延期し、前記データを記憶する手段の処理中は、前記汎用データ処理ユニットが前記受信専用領域からデータを読み出す処理を延期する調停手段を備えているデータ処理装置。

【請求項１８】　請求項１３において、前記通信手段は、他の前記データ処理装置からデータを要求されると前記データメモリの所定のアドレスからデータを提供する手段を備えているデータ処理装置。

【請求項１９】　請求項１８において、前記通信手段は、前記データを提供

する手段がデータを取得する前記データメモリの送信専用領域が前記汎用データ処理ユニットにより書き込まれているときは、前記データを提供する手段の処理を延期し、前記データを提供する手段の処理中は、前記汎用データ処理ユニットが前記送信専用領域にデータを書き込む処理を延期する調停手段を備えているデータ処理装置。

【請求項２０】　専用命令により実行される特定のデータ処理用のデータパス部を備えた少なくとも１つの専用データ処理ユニットと、汎用命令により汎用処理を実行可能な汎用データ処理ユニットと、前記専用命令および汎用命令を備えたプログラムに基づき、前記専用データ処理ユニットおよび汎用データ処理ユニットに命令を発行する命令発行ユニットと、前記プログラムを記憶するコードメモリと、前記汎用命令によりデータを入力または出力可能なデータメモリとを有するデータ処理装置の制御方法であって、

前記汎用命令に基づき入力または出力するデータの入力アドレスまたは出力アドレスが予め設定されたアドレスのときに、他の前記データ処理装置との間でデータを交換する通信工程を有するデータ処理装置の制御方法。

【請求項２１】　請求項２０において、前記通信工程は、前記出力アドレスが前記予め設定されたアドレスのときに、他の前記データ処理装置へデータを送信する工程を備えているデータ処理装置の制御方法。

【請求項２２】　請求項２０において、前記通信工程は、前記入力アドレスが前記予め設定されたアドレスのときに、他の前記データ処理装置からデータを受信する工程を備えているデータ処理装置の制御方法。

【請求項２３】　請求項２０において、前記通信工程は、他の前記データ処理装置からデータを受信すると前記データメモリの所定のアドレスにデータを記憶する工程を備えているデータ処理装置の制御方法。

【請求項２４】　請求項２３において、前記通信工程では、前記データメモリの受信専用領域が前記汎用データ処理ユニットにより読み出されているときは、前記データを記憶する工程を延期し、前記データを記憶する工程中は、前記汎用データ処理ユニットが前記受信専用領域からデータを読み出す処理を延期するデータ処理装置の制御方法。

【請求項２５】　請求項２０において、前記通信工程は、他の前記第１のタイプのデータ処理装置からデータを要求されると前記データメモリの所定のアドレスからデータを提供する工程を備えているデータ処理装置の制御方法。

【請求項２６】　請求項２５において、前記通信工程では、前記データメモリの送信専用領域が前記汎用データ処理ユニットにより書き込まれているときは、前記データを提供する工程を延期し、前記データを提供する工程中は、前記汎用データ処理ユニットが前記送信専用領域にデータを書き込む処理を延期するデータ処理装置の制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ハードウェアで演算処理を実行可能なデータパスを備えた専用データ処理ユニットを備えたデータ処理装置およびそれを用いたデータ処理システムに関するものである。

【０００２】

【従来の技術】

ＬＳＩの大規模・微細化は数十年に及ぶ進展を遂げ、近年では極めて大きな機能を持ったシステムがシステムＬＳＩなどとしてシリコン上に具現化できるようになりつつある。このような背景にあって、インテル社のペンティアムＬＳＩに代表されるような高速・高性能な汎用ＬＳＩとは別に、目的に応じて性能を最大限に引き出す専用目的のシステムＬＳＩや、その応用分野では汎用ＬＳＩよりコストパフォーマンスの優れた解を引き出すシステムＬＳＩの需要が拡大している。例えば、携帯電話に見られるような低消費電力を要求されるＬＳＩや、ネットワーク機器に見られるようなリアルタイム応答性とデータもしくはパケット転送に適したＬＳＩ、さらには、画像データの転送を目的とした画像圧縮伸長に適したＬＳＩなどの通信ネットワーク応用分野とディジタルＴＶに代表される情報家電応用分野で特に顕著である。

【０００３】

【発明が解決しようとする課題】

このような要求に対し、専用目的のシステムＬＳＩを構築するにあたり、専用目的のプロセッサを開発および製造する方式が採用されつつある。大規模な専用システムＬＳＩが要求される場合、システムＬＳＩの機能、すなわち仕様は何らかの形式で高級言語（Ｃ言語やＪａｖａ言語）で定義され記述される。したがって、その高級言語を実行できるコンパイラ等の環境を備えたプロセッサ、あるいはそのような環境に対応できるプロセッサであることが要求される。このため、目的に応じた専用命令を装備させた専用プロセッサであれば、その高級言語で記述された内容を処理するための専用回路を備えているので極めてコストパフォーマンスの良いシステムＬＳＩを提供できる。

【０００４】

一方、処理速度を向上する手法としてマルチプロセッサにより並列処理を行う手法が知られている。したがって、Ｃ言語により記述された一つのプログラムを分割して複数の処理プロセスにし、これらの処理プロセスを並列に実行することができれば処理速度は大幅に向上する。そして、汎用プロセッサでは装備されにくい特殊な演算を処理する命令による演算処理は、汎用プロセッサではクロック数を多く費やし易いので、専用命令として専用データ処理回路で処理する設計とし、それらの専用データ処理回路により並列処理することにより大幅に処理速度が向上する。

【０００５】

しかしながら、Ｃ言語で記述されたシステムを複数の処理あるいはプロセスに分解して、それを処理可能な専用回路が設計できたとしても、それらの処理あるいはプロセスが並列に実行されるように制御するためには、相互の専用回路の処理状況などを伝達する何らかの通信機能も専用回路に設ける必要がある。さらに、その通信結果に基づき専用回路における処理を制御する機能も盛り込む必要がある。さらに、適用分野に応じて多種多様の演算が要求されるので、それぞれの演算に対応した専用回路を開発することに加えて、専用回路同士を並列に動作させるための機能が必要である。したがって、専用回路を並列動作させるシステムＬＳＩは、処理速度は大幅に向上すると考えられるが、そのようなシステムＬＳＩを設計し検査するためには膨大な時間とコストが必要となる。したがって、需

要に応じてタイムリーに提供することは難しく、さらに、コストパフォーマンスも悪く、これらを改善することができる技術は開示されていない。

【０００６】

そこで、本発明においては、複数の専用回路を並列動作させることが可能なシステムＬＳＩを短期間で低コストで開発し、提供することができるデータ処理システムおよびデータ処理装置を提供することを目的としている。そして、Ｃ言語やＪＡＶＡ言語などの高級言語で記載されたプロセスを複数のプロセスに分散して並列に実行することができるシステムＬＳＩを短期間で経済的に供給することができるデータ処理システムおよびデータ処理装置を提供することを目的としている。

【０００７】

さらに、ハードウェアを直接意識しないで記述できるＣ言語やＪＡＶＡ言語に対応した、あるいは連動した通信機能を採用することにより、Ｃ言語などで記述された大規模なシステムを、複数の専用回路を備えたシステムとして短期間に低コストで提供することができるデータ処理システムおよびデータ処理装置を提供することも本発明の目的としている。

【０００８】

【課題を解決するための手段】

本願の出願人は、たとえば、特開２０００－２０７２０２号にカスタマイズ可能な専用命令を装着できるデータ処理装置を開示している。このデータ処理装置は、専用データ処理ユニットであるＶＵ部と、汎用なデータ処理が可能なＲＩＳＣプロセッサに当るＰＵ部とを有するデータ処理装置であり、ＶＵ部はＰＵ部とは異なりマルチサイクルで動作可能であり、専用命令により大規模な処理を実行可能になっている。したがって、専用命令により実行される特定のデータ処理用のデータパス部、すなわち専用回路を備えた専用データ処理ユニットと、汎用データ処理ユニットとが組み合わされたデータ処理装置を組み合わせ、汎用データ処理ユニットに他のデータ処理装置の汎用データ処理ユニットとの通信機能を搭載することにより、複数の専用回路を備えたシステムであって、それらの専用回路を並列動作させることができるデータ処理システムを短期間に、そして経済的

に提供することが可能となる。さらに、Ｃ言語あるいはＪＡＶＡ言語で記述されたシステムのプログラムファンクションそのものを専用命令として一命令化して専用データ処理ユニットで実行可能とすることにより、Ｃ言語で記述されたシステムを複数の処理あるいはプロセスに分解し、それらの処理あるいはプロセスを専用回路で高速に並列に実行することができる処理能力の高いデータ処理システムを短期間に、そして低コストで提供することができる。

【０００９】

すなわち、本発明のデータ処理システムは、複数のデータ処理装置を有し、それらのうち２以上のデータ処理装置は、専用命令により実行される特定のデータ処理用のデータパス部を備えた少なくとも１つの専用データ処理ユニットと、汎用命令により汎用処理を実行可能な汎用データ処理ユニットと、専用命令および汎用命令を備えたプログラムに基づき、専用データ処理ユニットおよび汎用データ処理ユニットに命令を発行する命令発行ユニットとを有する第１のタイプのデータ処理装置であり、少なくとも１つの第１のタイプのデータ処理装置の汎用データ処理ユニットは、他の第１のタイプのデータ処理装置の汎用データ処理ユニットとデータを交換可能な通信手段を備えている。したがって、本発明のデータ処理装置は、専用命令により実行される特定のデータ処理用のデータパス部を備えた少なくとも１つの専用データ処理ユニットと、汎用命令により汎用処理を実行可能な汎用データ処理ユニットと、専用命令および汎用命令を備えたプログラムに基づき、専用データ処理ユニットおよび汎用データ処理ユニットに命令を発行する命令発行ユニットとを有し、汎用データ処理ユニットは、他のデータ処理装置の汎用データ処理ユニットとデータを交換可能な通信手段を備えている。

【００１０】

本発明の専用データ処理ユニットは、アプリケーションなどに特化した専用回路となるデータパス部を備えており、専用命令により特化した処理あるいはプロセスを高速で実行できる。一方、汎用データ処理ユニットは、専用命令に対処する必要がなく、基本命令あるいは汎用命令を解釈して実行できる機能があればよく、汎用性を犠牲にすることなく様々なアプリケーションなどに対応した専用データ処理ユニットと共存できる。そして、専用命令および汎用命令を備えたプロ

グラムに基づき、専用データ処理ユニットおよび汎用データ処理ユニットが制御されるので、汎用データ処理ユニットにより専用データ処理ユニットを制御したり、その演算結果による汎用的な処理を行うことができる。したがって、汎用データ処理ユニットに並列処理に必要な通信手段を設けることにより、通信機能を専用回路から分離して組み込むことが可能となり、さらに通信機能をプログラムにより制御することができる。

【００１１】

このため、複数の専用回路を有するデータ処理システムにおいて、それらの専用回路を並列に実行するために必要な通信機能を、専用回路に影響を与えず、汎用的な構成で簡単に設けることが可能であり、さらに、プログラムにより柔軟に制御できる。このため、複数の専用回路を並列実行可能なデータ処理システムの設計および開発期間を短縮でき、低コストで提供できる。さらに、通信機能はプログラムで制御が可能なので、後の変更や修正にも柔軟に対処できる。したがって、複数のデータ処理装置の専用データ処理ユニットを含む、単一のデータの流れを処理するデータ処理系統を形成したり、複数のデータ処理装置の専用データ処理ユニットを含む、データの流れを処理する複数のデータ処理系統を形成することにより、Ｃ言語やＪＡＶＡ言語などの高級言語で記載されたプロセスを複数のプロセスに分散して並列に実行できるシステムＬＳＩとして適したデータ処理システムおよびデータ処理装置を提供できる。

【００１２】

システム全体がＣ言語などの高級言語で記述されており、これを複数のプロセスに分割し、各々のプロセスを本発明のデータ処理装置に割り当てる場合においては、データ処理装置の間のデータ通信をいかにして行うかという課題に直面する。プロセッサ間のデータ通信はバスを通じて行う方式や、専用の通信専用ハードウェアマクロを介して行う方式が多く用いられている。したがって、本発明のデータ処理システムにおいて、データ処理装置の間の通信手段として、これらの通信専用のハードウェアを用いることも可能である。しかしながら、Ｃ言語を記述するユーザから直にデータ転送を制御もしくは管理し難いという欠点がある。すなわち、バス方式ではハードウェアであるバス自体をＣ言語のレベルから直接

参照するのが困難である。上述したように、Ｃ言語のような高級言語ではハードウェアを直接意識しないで記述できることに特徴があり、当然の事とも言える。また、通信専用ハードウェアマクロを使用してデータ通信を行う場合には、通信機能は専用ハードウェアが保持するのでＣ言語のレベルからでは細かな制御もしくはプログラミングが困難ということになる。すなわち、これらの従来多く採用されているプロセッサ間データ通信機構は、ハードウェア側の要求に基づいてボトムアップに構築されたものである。従って、Ｃ言語との連動性は要求されておらず、そのため連動性は薄いといえる。

【００１３】

しかしながら、本発明のデータ処理システムに基づき、Ｃ言語で記述された仕様に基づいてシステムＬＳＩを設計するためには、Ｃ言語で記述されたシステム仕様からＬＳＩ化に向けてトップダウンな設計方式であることが望ましい。したがって、Ｃ言語でハードウェアを意識することなしにデータの転送が自由に行えることが望ましい。すなわち、本発明のデータ処理システムにおいては、Ｃ言語で記述されたシステム全体を複数のＣ言語のプロセスに分割し、それらの各々の対応した専用回路を備えた複数のデータ処理装置の集合としてシステムＬＳＩを設計することができる。そして、複数のＣ言語のプロセスに分解する際に、Ｃ言語により、ハードウェアを意識すること無しにデータを転送する記述が可能となれば、複数のＣ言語のプロセスに分解する設計作業を円滑に進めることがはじめて可能となる。そのためには、Ｃ言語により、ハードウェアを意識すること無しにデータ転送を自由に行えるハードウェア・アーキテクチャを提供する必要がある。

【００１４】

このため、本発明においては、汎用命令に基づくデータの入出力において、そのデータを入力または出力するアドレスにより、データを他のデータ処理装置のデータメモリから入力、または他のデータ処理装置のデータメモリに対し出力するようにしている。すなわち、本発明のデータ処理装置は、プログラムを記憶するコードメモリ、たとえば、メモリのプログラム記憶領域、コードＲＡＭあるいはコードＲＯＭと、汎用命令によりデータを入力または出力可能なデータメモリ

、たとえば、メモリのデータ記憶領域あるいはデータＲＡＭとを有しており、通信手段は、汎用命令に基づき入力または出力するデータの入力アドレスまたは出力アドレスが予め設定されたアドレスのときに、他のデータ処理装置との間でデータを交換し、他のデータ処理装置のデータメモリからデータを入力し、またはそのデータメモリに対しデータを出力するようにしている。また、本発明の、データ処理装置の制御方法においては、汎用命令に基づき入力または出力するデータの入力アドレスまたは出力アドレスが予め設定されたアドレスのときに、他のデータ処理装置との間でデータを交換する通信工程を有する。

【００１５】

他のデータ処理装置のデータメモリに対しデータを入出力するタイプのデータ通信において、相手側となる他のデータ処理装置のデータメモリに書き込みに行くＰＵＴ型と、相手側となる他のデータメモリに読み出しにゆくＧＥＴ型とを提示することができる。そして、どちらもＣ言語からのデータ転送を制御することができる。すなわち、ＰＵＴ型のデータ処理装置の通信手段あるいは通信工程では、出力アドレスが予め設定されたアドレスのときに、他のデータ処理装置へデータを送信する。したがって、受信側となる他のデータ処理装置のデータメモリの少なくとも１部の領域を、自己のデータ処理装置のデータメモリと同じレベルで仮想的に取り扱うことができる。このため、Ｃ言語により、データの出力先を所定のアドレスにすると、他のデータ処理装置のデータメモリにデータを書き込むことができる。

【００１６】

一方、ＰＵＴ型のデータ処理装置の相手となる受信側のデータ処理装置の通信手段あるいは通信工程では、発信側の他のデータ処理装置からデータを受信するとデータメモリの所定のアドレスにデータを記憶する。これにより、受信したデータを自己のデータメモリに記憶できる。したがって、Ｃ言語によりデータが書き込まれたアドレスのデータを読み込むことにより、汎用データ処理ユニットではそのデータを使用できる。この結果、Ｃ言語により発信側と受信側のデータ処理装置間でデータを転送する処理を操作できたことになる。

【００１７】

たとえば、あるアドレスを予め設定しておき、そのアドレス以上であれば、他のデータ処理装置のデータメモリへ書き込み、そのアドレス以下であれば自身のデータメモリへ書き込む、という制御を行う。この制御を行う為に、通信相手となるデータ処理装置の情報を格納するレジスタを設け、そこに送出先のデータ処理装置の識別情報、そのデータ処理装置に対しデータ転送を開始するアドレス、転送を終了するアドレスなどの情報を格納しておくことができる。

【００１８】

同様に、ＧＥＴ型のデータ処理装置の通信手段あるいは通信工程では、入力アドレスが予め設定されたアドレスのときに、他のデータ処理装置からデータを受信する。したがって、送信側となる他のデータ処理装置のデータメモリの少なくとも１部の領域を、自己のデータ処理装置のデータメモリと同じレベルで仮想的に取り扱うことができる。このため、Ｃ言語により、データの入力元を所定のアドレスにすると、他のデータ処理装置のデータメモリからデータを読み込むことができる。

【００１９】

一方、ＧＥＴ型のデータ処理装置の相手となる送信側のデータ処理装置の通信手段あるいは通信工程では、受信側となる他のデータ処理装置からデータを要求されるとデータメモリの所定のアドレスからデータを提供する。これにより、Ｃ言語によりデータをデータメモリの所定のアドレスに書き込むことにより、受信側のデータ処理装置にデータを転送できたことになる。このようにＧＥＴ型においても、Ｃ言語により発信側と受信側のデータ処理装置間でデータを転送する処理を操作できたことになる。

【００２０】

データが誤り無く転送されるには、転送するデータを入力または出力する領域に、送信側および受信側のデータ処理装置が同時に入力または出力しないようにすることが望ましい。本発明のデータ処理装置は、データを転送するタイミングをプログラムで制御することができるので、そのような事態が発生しないように送信側および受信側のデータ処理装置のプログラムを作成でき、Ｃ言語により制御することができる。また、通信手段に、データを記憶する手段がデータを記憶

するデータメモリの受信専用領域が汎用データ処理ユニットにより読み出されているときは、データを記憶する手段の処理を延期し、データを記憶する手段の処理中は、汎用データ処理ユニットが受信専用領域からデータを読み出す処理を延期する調停手段、または、データを提供する手段がデータを取得するデータメモリの送信専用領域が汎用データ処理ユニットにより書き込まれているときは、データを提供する手段の処理を延期し、データを提供する手段の処理中は、汎用データ処理ユニットが送信専用領域にデータを書き込む処理を延期する調停手段を設けても良い。また、本発明のデータ処理装置の制御方法の通信工程で調停手段と同様の制御を行うようにしても良い。

【００２１】

このように、本発明は、専用データ処理ユニットと、通信手段を備えた汎用データ処理ユニットとを有するデータ処理装置を複数有するデータ処理システムを提供するものであり、本発明のデータ処理システムにより、複数の専用回路を並列実行することができるシステムＬＳＩを極めて短期間に、そして低コストで提供することができる。さらに、本発明においては、専用回路を備えた分散処理システムであるデータ処理装置の間の通信機構を、Ｃ言語あるいはＪＡＶＡ言語などの高級言語と連動性および対応性のあるハードウェアで実現するアーキテクチャを提供しており、１のプロセスから他のプロセスへデータ転送がＣ言語で記述でき、その結果、Ｃ言語によるプロセスを複数のプロセスに分割が容易となり、分散処理システムの設計が可能となる。したがって、Ｃ言語で記述された仕様を実現し、高速で処理可能な複数の専用回路を用いた分散処理タイプのシステムＬＳＩをさらに短期間に、経済的に設計し供給することができる。

【００２２】

【発明の実施の形態】

以下に図面を参照しながら本発明についてさらに説明する。図１に、特定の処理に特化した専用データ処理ユニット（専用命令実行ユニット、以降ではＶＵ）１と、汎用的な構成の汎用データ処理ユニット（汎用命令実行ユニットあるいはプロセスユニット、以降ではＰＵ）２とを備えた本発明のデータ処理装置１０の概要を説明する。このデータ処理装置１０は、専用回路を備えたプログラマブル

なプロセッサであり、このため、実行形式の制御プログラム（プログラムコード、マイクロプログラムコード）４ａを内蔵したコードＲＡＭ４から命令をフェッチし、専用データ処理ユニット１および汎用データ処理ユニット２にデコードされた制御信号を提供するするフェッチユニット５を備えている。本例においては、このフェッチユニットＦＵ５が命令発行ユニットに該当する。

【００２３】

このフェッチユニット５は、前の命令あるいはステートレジスタ６の状態、割り込み信号φｉなどによって決まる所定のコードＲＡＭ４の所定のアドレスから命令をフェッチするフェッチ部７と、フェッチされた専用命令あるいは汎用命令（一般命令）をデコードするデコード部８とを備えている。デコード部８は、専用命令をデコードした制御信号（デコーデド・コントロール・シグナル；Decoded Control Signal）φｖおよび汎用命令をデコードした制御信号（デコーデド・コントロール・シグナル；Decoded Control Signal）φｐを、専用データ処理ユニットＶＵ１および汎用データ処理ユニットＰＵ２にそれぞれ供給する。さらに、ＰＵ２からは実行状態を示すステータス信号（Exec unit Status Signal）φｓが返され、ＰＵ２およびＶＵ１の状態がステートレジスタ(状態レジスタ)６に反映されるようになっている。

【００２４】

本例のＰＵ２は、汎用レジスタ、フラグレジスタおよび演算ユニット（ＡＬＵ）などから構成される汎用性の高い実行ユニット１１と、他のＰＵ２との間でデータを交換する機能を備えた通信ユニット１２とを備えており、データＲＡＭ１５を一時的な記憶領域としてデータを入出力しながら汎用処理を実行できるようになっている。これらのフェッチユニットＦＵ５、汎用データ処理ユニットＰＵ２、コードＲＡＭ４、データＲＡＭ１５を有する構成は、個々の機能は異なるが一般的なプロセッサユニットと類似の構成となる。したがって、ＦＵ５、ＰＵ２、コードＲＡＭ４およびデータＲＡＭ１５を有する構成をプロセッサユニット３と称することも可能であり、プロセッサユニット（ＰＵＸ）３からＶＵ１を制御するような概念で本例のデータ処理装置１０を構成あるいは設計することができる。

【００２５】

ＦＵ５からの専用命令φｖを実行する専用データ処理ユニットＶＵ１は、ＦＵ５が供給する命令がＶ命令φｖであるかなどをデコードするユニット２２と、予め特定のデータ処理を行うように制御信号をハードウェア的に出力するシーケンサ（ＦＳＭ（Finite State Machine）、ファイナイトステートマシン）２１と、このシーケンサ２１からの制御信号に従って特定のデータ処理を行うようにデザインされたデータパス部２０を備えている。また、ＶＵ１は、ＰＵ２からアクセス可能なレジスタ２３を備えており、データパス部２０の処理に必要なデータをインターフェイスレジスタ２３を介してＰＵ２で制御したり、ＶＵ１の内部状態をレジスタ２３を介してＰＵ２で参照できるようになっている。また、データパス部２０で処理された結果はＰＵ２に供給され、ＰＵ２ではその結果を利用した処理が行われる。

【００２６】

本例のデータ処理装置１０は、コードＲＡＭ４に、汎用命令（Ｐ命令）および専用命令（Ｖ命令）を含んだプログラムが記憶されており、それがフェッチユニット５でフェッチされ、デコードされた制御信号φｐまたはφｖとしてＶＵ１およびＰＵ２に供給される。ＶＵ１は、制御信号φｐおよびφｖのうち、自己を起動する専用命令の制御信号φｖが供給されると稼動する。一方、ＰＵ２には、汎用命令がデコードされた制御信号φｐだけが供給されるようになっており、Ｖ命令をデコードした制御信号φｖはＰＵ２には発行されず、その代わりに、実行を伴わないｎｏｐ命令を示す制御信号が発行され、ＰＵ２の処理はスキップされる。ＶＵ１は、アプリケーションなどによって変更されるものであり、ＶＵ１に指示を出す専用命令もアプリケーションによって変わることが多い。ＶＵ１は、アプリケーションに特化した専用回路であり、Ｖ命令をデコードした制御信号を解釈するように設計することは容易である。一方、ＰＵ２は、ｎｏｐ命令が出力されることにより、ＶＵ１用に特化した命令に対処する必要がなく、基本命令あるいは汎用命令を解釈して実行できる機能があればよく、汎用性を犠牲にすることなく様々なアプリケーションに対応したＶＵ１と共存し、これらを制御したり、その演算結果を用いて処理を行うことができる。

【００２７】

このように、図１に示したデータ処理装置１０は、リアルタイム応答などの特殊な演算が要求される処理を実現できる専用回路を備えたＶＵ１と、汎用性があるＰＵ２とを有するものであり、以降においてはＶＵＰＵと称することにする。このＶＵＰＵ１０は、リアルタイム応答性を犠牲にすることなく、設計および開発期間を短縮でき、さらに、その後の変更や修正にも柔軟に対処できるものである。また、ＶＵ１は、１つに限定されることはなく、アプリケーションで要求される専用処理を処理できるように複数のＶＵ１を用意し、それぞれのＶＵ１を稼動する複数の専用命令をプログラムコードに含めることが可能である。さらに、本例のＶＵ１は、特殊な演算処理だけでなく、プログラム中の特定のプログラムファンクションを専用回路化してプログラムを効率良く可動させることができる。そして、本例のＰＵ２は、他のＰＵ２とデータを交換することができる通信ユニット１２を備えており、他のＶＵＰＵ１０と通信することにより、複数のＶＵＰＵ１０のＶＵ１を並列に稼動させることができる。したがって、ＶＵＰＵ１０を複数備えたデータ処理システムは適応可能な範囲が非常に広いアーキテクチャである。

【００２８】

たとえば、図２に示すように、Ｃ言語により記述されたプロセスが図２のように、親となるプロセスＣ１と、そのプロセスＣ１からデータが転送され、そのデータに基づく処理結果を返すプロセスＣ２およびＣ３により構成されている場合、これらのプロセスＣ１、Ｃ２およびＣ３を図３に示すように３つのＶＵＰＵ１０に割り振ることが可能である。そして、ＶＵＰＵ１０であれば、特殊な演算処理だけでなく、プログラム中の特定のプロセスあるいはプログラムファンクションを専用回路化してプログラムを効率良く可動させることができるので、処理速度を向上できる。さらに、ＶＵＰＵ１０は、ＰＵ２が通信機能を備えているので、図４に示すように、親となるプロセスＣ１が割り振られたＶＵ１、すなわちＶＵ（Ｃ１）を備えたＶＵＰＵ１０から、子供となるプロセスＣ２が割り振られたＶＵ（Ｃ２）を備えたＶＵＰＵ１０に対しデータが転送されることにより、ＶＵ（Ｃ２）がＶＵ（Ｃ１）と並列に処理を開始することができる。そして、ＶＵ（

Ｃ２）の処理結果をＶＵ（Ｃ１）に返すことによりＶＵ（Ｃ１）ではその処理結果に基づく処理を実行することができる。

【００２９】

同様に、プロセスＣ３が割り当てられたＶＵ（Ｃ３）を備えたＶＵＰＵ１０に対しデータが転送されることにより、ＶＵ（Ｃ３）がＶＵ（Ｃ１）と並列に処理を開始することができる。さらに、ＶＵ（Ｃ２）とＶＵ（Ｃ３）とが並列に処理を行うことができるプロセスであればさらに並列度を上げることが可能であり、処理速度を向上することができる。このように、各ＶＵＰＵ１０がある時刻でひとつしか動作しなければ、非並列であり、元のＣ言語で記述されたプロセスを専用回路化した効果しか得られない。これに対し、本発明のＶＵＰＵ１０であれば、専用回路化した複数の複数のプロセスを並列に実行することが可能となり、処理速度を大幅に向上できる。このため、図３に示すように、Ｃ言語で記述された仕様を複数のプロセスに分け、複数のＶＵＰＵ１０のＶＵに割り当て、それらのＶＵＰＵ１０によってシステムＬＳＩ３０などのデータ処理システムを構築することにより、プロセスあるいはファンクションを専用回路化するメリットに加えて、それらの専用回路を並列に実行することが可能となる。したがって、非常に処理速度の速いシステムＬＳＩ３０を提供することが可能となる。

【００３０】

すなわち、図５に示すように、Ｃ言語で記述されたある仕様５１が与えられたときに、その仕様５１をある程度並列実行可能な複数のプロセス５２に分けることが可能である。そして、専用回路を形成するデータパス部２０とシーケンサ２１により、プロセス５２の全てあるいは一部が実行できるようにＶＵ１を生成することが可能であり、ＶＵＰＵ１０として供給することができる。そして、そのようにして作成されたＶＵＰＵ１０を組み合わせてシステムＬＳＩ３０とすることにより、並列度の高い処理が可能なシステムＬＳＩ３０を提供することができる。さらに、ＶＵＰＵ１０は、専用回路で処理することが適さない処理はプロセッサとしての機能を備えたＰＵ２により処理することが可能であり、専用回路による処理を並列に実行できるのみならず、汎用プロセッサによる処理も並列に実行させることができる。

【００３１】

図６ないし図８は、通信機能を備えた本発明のＶＵＰＵ１０によりデータ処理システム３０を構成する幾つかの例を示してある。多くのケースでは、１つのチップに複数のＶＵＰＵ１０が搭載され、本明細書に示したような構成のデータ処理システム３０は、特定のアプリケーションの処理を効率良く行うことができるシステムＬＳＩとして提供されることになるであろう。図６に示したデータ処理システム３０は、ＶＵＰＵ１０のＰＵ２と通信するのに適したアーキテクチャを備えたプロセッサ３１を中心に、複数のＶＵＰＵ１０が適当な通信手段により接続されているものである。たとえば、並列に稼動する複数のＶＵ１により、画像データとなるビットストリーム３９に対し圧縮あるいは解凍に必要な一連の処理を順次施すことが可能であり、画像処理を高速に実行することができる。そして、各処理を行うＶＵ１はＰＵ２で制御され、そのＰＵ２は他のＰＵ２とデータを交換できるようになっているので、処理の同期、調停あるいはエラーなどを適切に処理することができる。これらのＶＵＰＵ１０は、各々が独立したプログラムコードにより動作するので、マルチインストラクションによりシングルデータフローを処理することができるデータ処理システム３０を提供することができる。

【００３２】

図７に示したデータ処理システム３０は、汎用のバスなどを介してデータを送受信することができる通信機能を備えたＶＵ（ＣＯＭ）を搭載したＶＵＰＵ１０Ａをインターフェイスとして用い、ＶＵＰＵ１０を繋げたシステムと、ＶＵＰＵとは異なるアーキテクチャの従来あるいは他のタイプのプロセッサ３２とにより構築されている。また、図８に示したデータ処理システム３０は、ＶＵ（ＣＯＭ）とプロセスを搭載したＶＵ（Ｃ１）あるいはＶＵ（Ｃ２）の２つのＶＵを搭載したＶＵＰＵ１０Ｂをインターフェイスとして、他のタイプのプロセッサ３２を含めてシステムを構築した例である。このように、通信機能を備えたＰＵ２を採用することにより、複数のＶＵＰＵ１０を用いたシステムを非常にフレキシブルに構築することが可能であり、様々な仕様のアプリケーションに対しの最適な構成のシステムＬＳＩを提供できる。

【００３３】

このように、複数のＶＵＰＵ１０を並列実行させることにより極めて処理速度の速いシステムＬＳＩを提供することが可能となる。そのためには、図９に示すようにＣ言語で記述された機能あるいは仕様５１を複数のプロセス５２に分解してＶＵＰＵ１０を作成する必要があるが、ＶＵＰＵ１０の間のデータ通信をいかにして行うかという課題に直面する。プロセッサ間のデータ通信はバスを通じて行う方式や、専用の通信専用ハードウェアマクロを介して行う方式が多く用いられており、本例のデータ処理システム３０にも適用できる。

【００３４】

しかしながら、たとえば、バス方式ではハードウェアであるバス自体をＣ言語のレベルから直接参照するのが困難であり、Ｃ言語により複数のプロセス５２に分解したときに通信機能をＣ言語のレベルからでは細かな制御ができない。したがって、上述したような複数のＶＵＰＵ１０を備えたデータ処理システムを短期間に低コストで開発するためには、Ｃ言語でハードウェアを意識することなしにデータの転送が自由に行えることが望ましい。すなわち、複数のＣ言語のプロセスに分解する際に、Ｃ言語により、ハードウェアを意識すること無しにデータを転送する記述が可能となれば、複数のＣ言語のプロセスに分解する設計作業を円滑に進めることがはじめて可能となる。そして、Ｃ言語のレベルで分割されたプロセスに基づき、専用回路化できる部分をＲＴＬに変換して専用回路を設計および製造し、専用回路を稼動する専用命令とその他の汎用処理を行う汎用命令を備えたプログラムコードを作成し、さらにこれらをテストして完成するステップ５３の負荷を軽減することができる。

【００３５】

このため、本例ではＣ言語により、ハードウェアを意識すること無しにデータ転送を自由に行えるハードウェア・アーキテクチャを通信機能として採用している。この方式の通信機能は、Ｃ言語に限定されるものではなく、より分散および並列記述の容易なＪＡＶＡ言語、あるいはその他の高級言語で記載された仕様をシステムＬＳＩなどのデータ処理システムとして実現するためにも好適である。プロセスを複数のプロセスに分散して並列に実行できるシステムＬＳＩとして適したデータ処理システムおよびデータ処理装置を提供できる。

【００３６】

図１０に、本例のＶＵＰＵ１０の構成をＰＵ２を中心に示してある。ＰＵ２は、図１に基づき説明したように、コードＲＡＭ４に格納されたプログラム４ａの汎用命令をデコードした制御信号φｐを実行する実行ユニット１１と、通信機能を備えた通信ユニット１２とを備えている。そして、本例の通信ユニット１２は、実行ユニット１１がデータＲＡＭ１５にアクセスするために出力するアドレスＡＯが予め設定された範囲のアドレスであるときは、通常のＲＤ／ＷＲデータＲＡＭ１５Ｎとは異なる受信用データＲＡＭ１５Ｘあるいは送信用データＲＡＭ１５Ｙに対し入出力動作を行う。そして、自己の受信用データＲＡＭ１５Ｘに書き込まれたデータを読み込んだり、他のＶＵＰＵの送信用データＲＡＭ１５Ｙからデータを取得することにより、他のＶＵＰＵとの間でデータを行う。すなわち、本例のＶＵＰＵ１０のプロセッサＰＵＸ３は、コードＲＡＭ４と、データＲＡＭ１５とが異なる、いわゆるハーバードアーキテクチャと称されるタイプである。そして、データＲＡＭの一部を他のＶＵＰＵ１０と共用したり、他のＶＵＰＵ１０と共用のデータＲＡＭを設けることにより、入出力アドレスにより他のＶＵＰＵ１０に対しデータ転送することができる。したがって、入出力アドレスをＣ言語で記述することによりにＶＵＰＵ１０の間の通信を制御することができる。

【００３７】

この通信方式は、通信相手のＶＵＰＵ１０の受信ＲＡＭ１０Ｘに出力データを書き込むＰＵＴ型と、通信相手のＶＵＰＵ１０の送信ＲＡＭ１０Ｙから入力データを取得するＧＥＴ型に大きく分かれる。図１０に示したＶＵＰＵ１０は、ＰＵＴ型の例である。したがって、ＶＵＰＵ１０は、入出力可能な通常のＲＤ／ＷＲデータＲＡＭ１５Ｎに加え、自己の実行ユニット１１に対してはリードオンリとなる受信ＲＡＭ（受信データＲＡＭ）１５Ｘを備えている。また、通信ユニット１２は、出力データＤＯを他のＶＵＰＵ１０に送信する送信インターフェイス１３と、他のＶＵＰＵ１０から受信した入力データＤＩを受信ＲＡＭ１５Ｘに書き込む受信インターフェイス１４を備えている。

【００３８】

送信インターフェイス１３は、送信制御部１３Ｃを備えており、実行ユニット

１１がプログラム４ａにしたがってデータを書き込む際に出力するアドレスＡＯがあるアドレス以上であると、送信バッファ１３Ｂを経由して他のＶＵＰＵのデータＲＡＭ（受信ＲＡＭ）に書き込む。したがって、プログラム４ａからみると、自身のＶＵＰＵ１０に実体のあるデータＲＡＭに書き込むのと同じ操作で、実体の無い送信用のデータＲＡＭ１５Ｚにデータを転送することができる。そして、その実体のない送信用のデータＲＡＭ１５Ｚは、通信相手のＶＵＰＵに存在する送信専用のライトオンリのデータＲＡＭ１５Ｘであり、通信相手の実行ユニット１１にとってはリードオンリの受信専用のデータＲＡＭとなる。

【００３９】

受信インターフェイス１４は、受信制御部１４Ｃを備えており、他のＶＵＰＵ１０から受信した入力データＤＩ（送信元においては出力データＤＯ）を受信ＲＡＭ１５Ｘに書き込む。送信制御部１３Ｃおよび受信制御部１４Ｃは、それぞれ、コンフィグレーション・レジスタ１３Ｒおよび１４Ｒを備えている。送信用のコンフィグレーション・レジスタ１３Ｒには、送信先のＶＵＰＵの識別情報（ＩＤ）、送信開始アドレス、転送サイズ、さらには送信終了アドレスなどの実行ユニット１１から出力されるデータを転送先に送信するために必要なデータが格納される。受信用のコンフィグレーション・レジスタ１４Ｒには、受信源となる送信元のＶＵＰＵのＩＤ、受信開始アドレスおよび受信終了アドレスなどのデータを受信するために必要なデータが格納される。また、送信元の実体の無い送信専用のデータＲＡＭ１５Ｚのアドレスと、送信先の受信専用のデータＲＡＭ１５Ｘの受信アドレスが一致しない場合には、それらのアドレスの対応表を送信側あるいは受信側のコンフィグレーション・レジスタ１３Ｒあるいは１４Ｒに登録しておき、送信時あるいは受信時にアドレス変換することができる。

【００４０】

これらの送信用のコンフィグレーション・レジスタ１３Ｒおよび受信用のコンフィグレーション１４Ｒの内容は、例えば、ＰＵ２の汎用レジスタ１１Ｒを介してプログラム４ａを通じて設定することができる。したがって、Ｃ言語により送信および受信を行うことになる入力および出力アドレスや、アドレス変換などの初期条件を設定することができる。

【００４１】

また、実行ユニット１１に入力されるデータＤＩは、受信用のコンフィグレーション・レジスタ１４Ｒに格納されているアドレスの内容から、受信専用のデータＲＡＭ１５Ｘからの読出か、通常のデータＲＡＭ１５Ｎからの読出かを判断することが可能である。このため、受信ＲＡＭ１５Ｘの出力ＤＯと、ＲＤ／ＷＲデータＲＡＭ１５Ｎの出力ＤＯは、受信制御回路からの信号により制御されるセレクタ１６を経由して実行ユニット１１のＤＩに供給される。この結果、プログラム４ａは、自己が入出力可能なデータＲＡＭ１５Ｎのデータか、通信相手が書き込んだ受信ＲＡＭ１５Ｘのデータかはアドレスの違いで制御することが可能であり、その他の操作は全く同等に処理できることとなる。

【００４２】

さらに、送信インターフェイス１３は、調停回路１３Ａを備えており、データ書込状態を示す信号φｐｕｔを送出する。それと共に、送信を開始するに当たっては、相手がデータを読出中でない事を確認する必要があり、送信相手のＶＵＰＵにおける受信ＲＡＭ１５Ｘのデータ読出状態を示す信号φｂｕｓｙによって識別する。したがって、データ読出状態を示す信号φｂｕｓｙは転送する相手のプロセッサの数分（ＩＤ分）だけ必要となる。受信インターフェイス１４も調停回路１４Ａを備えており、受信専用のデータＲＡＭ１５Ｘから読出中は、他のＶＵＰＵからの出力データを受信できないようになっている。そのため、データ書込状態を示す信号φｐｕｔを受信したときに、受信ＲＡＭ１５Ｘが読み出し中であると、読出状態を示す信号φｂｕｓｙを出力する。送信インターフェイス１３および受信インターフェイス１４で取り扱われるこれらの書込状態を示す信号φｐｕｔと読出状態を示す信号φｂｕｓｙは、方向は逆だが同じ種類に信号である。そして、レベル信号で送出されるのが一般的である。

【００４３】

本例の受信専用のデータＲＡＭ１５Ｘは、デュアルポートデータＲＡＭであるが、シングルポートデータＲＡＭにより構成することが可能である。デュアルポートデータＲＡＭであれば受信しながらの読出操作が可能となり、並列性が向上し、さらに、上記のような調停回路を省くことができる可能性がある。しかしな

がら、書込みアドレスＡＩと、読出しアドレスＲＡＩが同一の場合があることを考慮すると上記の調停回路１３Ａおよび１４Ａと状態信号φｐｕｔおよびφｂｕｓｙを設けておくことが望ましい。調停回路を省いた場合でも、書込みアドレスＡＩと、読出しアドレスＲＡＩが同一の場合を考慮すると、入力データＤＩから読出しデータＲＤＩにＲＡＭをバイパスしてデータを出力可能な論理回路が必要となる。

【００４４】

以上に示す送信・受信機構全体をＩＶＣ機構（Ｉｎｔｅｒ　ＶＵＰＵ　Ｃｏｍｍｕｎｉｃａｔｉｏｎ機構）と称することとする。

【００４５】

図１１に、ＩＶＣ機構を備えた２つのＶＵＰＵ１０の間でデータ交換される様子を各々のＰＵのメモリマップ１９を用いて示してある。本図から分かるように、ＰＵＴ型のＩＶＣ機構においては、アドレスがＡ１からＡ２の範囲であるときは、相手側のデータＲＡＭ１５Ｘにデータを書き込むことによりデータを転送する。したがって、データＲＡＭの使用効率が高く、また、データの二重持ちを防ぐことができるのでデータに齟齬が発生することも防止できる。また、アドレスがＡ３からＡ４のときは、相手側のＰＵによりデータが書き込まれたデータＲＡＭ１５Ｘからデータを取得する。したがって、転送されたデータをＰＵ２で使用して処理を進めることができる。

【００４６】

図１２に、ＰＵＴ型のＩＶＣ機構を備えた４つのＶＵＰＵ１０を接続したデータ処理システム３０の例を示してある。この例では、１つのＶＵＰＵ１０ｐが親となり、他の３つのＶＵＰＵ１０ｃは子供となっており、親のＶＵＰＵ１０ｃからは全ての子供のＶＵＰＵ１０ｐに対し同様にデータが転送され、子供のＶＵＰＵ１０ｃからは親ＶＵＰＵ１０ｐへ個別にデータが転送される。このため、親のＶＵＰＵ１０ｐは、子供の数に相当する受信ＲＡＭあるいは受信ＲＡＭ領域１５Ｘを備えており、子供のＶＵＰＵ１０ｃは１つの受信ＲＡＭあるいは受信ＲＡＭ領域１５Ｘを備えている。したがって、親のＶＵＰＵ１０ｐにおいては、子供のＶＵＰＵ１０ｃからデータを並列に受信することが可能であり、また、受信した

データを保持しておけるのでプログラムにより適当なときに使用できる。一方、親のＶＵＰＵ１０ｃの受信ＲＡＭ１５Ｘを１つにすることも可能であり、この場合は、順番に子供のＶＵＰＵ１０ｐからデータを受信するように、親のＶＵＰＵ１０ｐおよび子供のＶＵＰＵ１０ｃのプログラムを作成する必要がある。

【００４７】

また、本例のシステムでは、ＶＵＰＵ１０ｐおよびＶＵＰＵ１０ｃの間では、４本のデータ転送可能な経路を備えたチャネル３５が用意されている。このようなプロセッサ間のデータ転送経路は、一般的な信号通信処理により形成することが可能である。そして、チャネル数を増加させれば子供のＶＵＰＵ１０ｃ同士が直接通信する構成にすることも可能であり、本発明のＩＶＣ機構を備えたＶＵＰＵを用いて通信経路を構築するアーキテクチャは自由度が高い。

【００４８】

図１３は、図１２に示したデータ処理システム３０の各々のＶＵＰＵのＰＵにおけるメモリ構成を示してある。上記と同様に、ＰＵＴ型のＩＶＣ機構を備えたＶＵＰＵ１０を用いているので、１対Ｎのシステムであってもデータを送出する場合にはますますシステム全体の分散性を高め、かつ、データＲＡＭの使用効率を向上することができる。たとえば、親のＶＵＰＵ１０ｐのＰＵ（ＰＵ－Ａ）においては、メモリマップ１９の送信ＲＡＭの領域はＶＵＰＵ１０ｐには実体がなく、そのアドレスに物理的に対応するデータＲＡＭは、子供のＶＵＰＵ１０ｃにそれぞれ分散して配置されている。また、子供のＶＵＰＵ１０ｃのＰＵ（ＰＵ－Ｂ、ＰＵ－ＣおよびＰＵ－Ｄ）においても、メモリマップ１９の送信ＲＡＭ領域は実体がなく、それらのアドレスに物理的に対応するデータＲＡＭは、親のＶＵＰＵ１０ｐに配置されている。

【００４９】

図１４に、本例のＩＶＣ機構を実現する通信ユニット１２の動作をフローチャートで纏めてある。実際に通信を開始する前に、送信用のコンフィグレーション・レジスタ１３Ｒに、送信先のＶＵＰＵのＩＤ、送信するデータの開始アドレス（実体のない送信ＲＡＭに割り当てられたアドレス）、受信ＲＡＭ１５Ｘの開始アドレスなどを設定し、受信用のコンフィグレーション・レジスタ１４Ｒに、送

信元となるＶＵＰＵのＩＤ、送信されるデータの開始アドレス、受信ＲＡＭの開始アドレスなどを設定する処理が行われる。これらのコンフィグレーション・レジスタ１３Ｒおよび１４Ｒの設定は、Ｃ言語のレベルであればインライン・アセンブル記述により設定できる。また、この処理をファンクションとしてサブルーチン化しておくことも可能である。

【００５０】

そして、プログラムにしたがって入出力アドレスが出力されると、通信ユニット１２においては、まず、ステップ６１でデータの入出力アドレスを判断する。入出力データが通常のデータＲＡＭに割り当てられたアドレスでないときは、ステップ６２で、アドレスに基づき出力処理か入力処理かを判断する。入力の場合は、ステップ６３で受信ＲＡＭ１５Ｘに送信されたデータが書き込み中でないこと、すなわち、書込み状態信号φｐｕｔの書込み終了を待ち、ステップ６４で自己の受信ＲＡＭ１５Ｘからデータを読み出す。それと同時に読出し状態信号φｂｕｓｙを読出しにして書込みを禁止し、読出しが終了すると読出し状態信号φｂｕｓｙを終了状態にする。

【００５１】

一方、ステップ６２で出力の場合は、ステップ６５で読出し状態信号φｂｕｓｙが読出し終了になるのを待ち、ステップ６６で出力データ（アドレスとデータおよびそれらアドレスとデータが有効である事を示すライトイネーブル信号）を転送先のＶＵＰＵ１０に送信する。それと共に、書込み状態信号φｐｕｔを書き込み状態にして読み出しを禁止し、書込みが終了すると書込み状態信号φｐｕｔを終了状態にする。このように、入出力のアドレスによりデータを通信先のＶＵＰＵ１０のデータＲＡＭ１５Ｘにする制御方法を採用することにより、Ｃレベルの記述でデータの入出力アドレスを管理あるいは制御することだけで複数のＶＵＰＵ１０の間でデータを簡単に交換することができる。

【００５２】

図１５に、ＰＵ－ＡからＰＵ－Ｂの受信ＲＡＭ１５Ｘにデータを書き込む様子をタイミングチャートで示してある。サイクル１では、ＰＵ－Ｂの読出し状態信号φｂｕｓｙがオンになっているので、転送データは有効にならず書き込まれな

い。さらに、通常、読出し状態信号φｂｕｓｙがオフとなってから１サイクルあけて書込が行われる。このため、サイクル３に、ＰＵ－Ａの書込み状態信号φｐｕｔがオンとなり、転送データがアドレスＡ、データＤおよびライトイネーブルＷＥ込みで受信側のＰＵ－Ｂの受信専用データＲＡＭ１５Ｘに転送される。そして、書込み状態信号φｐｕｔが出力されている間に有効なデータが送信されると、それが受信データＲＡＭ１５Ｘに書き込まれる。この例では３サイクルと５サイクル目が有効データであることを示している。

【００５３】

本発明のＩＶＣ機構においては、図１４に示した処理を通信ユニット１２のファームウェアやゲートロジックで実装することも可能であるが、データ転送のすべてをＣレベルの記述で制御することが可能である。図１６（ａ）は、送信側のＰＵ－Ａの転送手順をＣ言語のレベルで記述した例であり、図１６（ｂ）は、受信側のＰＵ－Ｂの転送手順をＣ言語のレベルで記述した例である。ＰＵ－Ａのプログラム７１では、ステップ７１ａでコンフィグレーション・レジスタ１３Ｒに送信スタートアドレスが指定される。ついでステップ７１ｂで転送相手の受信ＲＡＭにデータを書込むための送信を開始する。この際、ステップ７１ｃに示すように、送信先の読出し状態信号φｂｕｓｙをチェックし、書込み状態信号φｐｕｔをオンにする処理をファンクション・コールによりサブルーチン化しておくことができる。信号のチェックおよび設定が済むと、ステップ７１ｄで、書き込むためのデータを送出する。そして、データの送出が終わるとステップ７１ｅで終了処理を行うが、ステップ７１ｆに示すように書込み状態信号φｐｕｔをオフにする処理などをサブルーチン化しておくことができる。

【００５４】

一方、ＰＵ－Ｂのプログラム７２では、ステップ７２ａでコンフィグレーション・レジスタ１４Ｒに受信スタートアドレスが指定される。ステップ７２ｂで受信ＲＡＭに書込まれた送信元からのデータを読み出す処理を開始する。この際、ステップ７２ｃに示すように、送信元の書込み状態信号φｐｕｔをチェックし、読出し状態信号φｂｕｓｙをオンにする処理をファンクション・コールによりサブルーチン化しておくことができる。信号のチェックおよび設定が済むと、ステ

ップ７２ｄで、転送されたデータを読出し、ステップ７２ｅで読出し終了処理を行う。ここでも、ステップ７２ｆに示すように読出し状態信号φｂｕｓｙをオフにする処理などをサブルーチン化しておくことができる。書込み状態信号φｐｕｔおよび読出し状態信号φｂｕｓｙをオン状態にしたり、その状態を確認するのはレジスタ操作となる。このため、上記のように、ファンクション・コールによりサブルーチン化しておき、別途アセンブラによりレジスタ設定を行う方法が適している。

【００５５】

このように、本発明のＩＶＣ機構による通信方法は、データの転送をすべてＣ言語のレベルの記述により操作できる。先に説明したように、Ｃ言語による仕様を複数のＣ言語によるプロセスに分解してＶＵＰＵ化する設計手法により、Ｃ言語による仕様を並列処理および分散処理することができるシステムＬＳＩを設計することが可能であり、この際、データのやりとりがＣ言語のレベルで直接記述できることにより、ＶＵＰＵ化するのが容易となる。したがって、本発明のＩＶＣ機構を採用することにより、Ｃ言語による仕様から、並列実行可能な複数の専用回路を備えたシステムＬＳＩを設計および製造する期間を大幅に短縮でき、低コストで提供することができる。

【００５６】

図１７は、データを送信するＰＵ－Ａと、データを受信するＰＵ－Ｂの間の状態情報伝達とそれを構成する信号線を示している。上記の例では、図１７（ａ）に示すように、読取状態信号φｂｕｓｙと、書込状態信号φｐｕｔの各々の専用の信号線に情報を持たせている。このため、図１７（ｂ）に示すように、それらの状態信号に対応する読取状態提示専用信号線７５と、書込状態提示専用信号線７６が、データを転送する信号線７７に加えて必要になる。

【００５７】

これに対し、状態情報の伝達に、受信データＲＡＭ１５Ｘを専用信号線に代わって使用する方法がある。上記の専用信号線を用いた方法では、アセンブラによるレジスタ操作を介してＣ言語のレベルから操作する必要があるのに対し、受信データＲＡＭ１５Ｘを使用すると、データに意味を持たせるので、すべてＣ言語

のレベルからデータ操作により転送処理を行うことができる。

【００５８】

図１８（ａ）に、送信側のＰＵ－Ａの転送手順をＣ言語のレベルで記述した例を示し、図１８（ｂ）は、受信側のＰＵ－Ｂの転送手順をＣ言語のレベルで記述した例を示してある。ＰＵ－Ａのプログラム７１では、ステップ７１ａでコンフィグレーション・レジスタ１３Ｒに送信スタートアドレスを指定すると共に、ステップ７１ｇで、自己の受信ＲＡＭ１５Ｘのアドレスで、受信側、すなわち、送信先の読取状態信号φｂｕｓｙが格納されるアドレスを指定する。送信先のＰＵ－Ｂが受信ＲＡＭ１５Ｘを読み出している状態のときは、送信元の受信ＲＡＭ１５Ｘの読取状態信号φｂｕｓｙが格納されるアドレスにフラグが立つ。したがって、転送相手の受信ＲＡＭにデータを書込むための送信を開始する際は、まず、ステップ７１ｈで、自己の受信ＲＡＭ１５Ｘの読取状態信号φｂｕｓｙが格納されるアドレスのデータを参照して送信先の状態をチェックする。ついで、ステップ７１ｉで、送信先の受信ＲＡＭ１５Ｘの受信スタートアドレスにフラグを立てて書込みを開始したことを伝達する。すなわち、本例では、受信スタートアドレスのデータが書込み状態信号φｐｕｔが格納されるアドレスとなっている。その後、ステップ７１ｊで書き込むためのデータを送出し、ステップ７１ｋで送信先の受信スタートアドレスにフラグをクリアするデータを送出し、書込みを終了する。

【００５９】

一方、ＰＵ－Ｂのプログラム７２では、ステップ７２ａでコンフィグレーション・レジスタ１４Ｒに受信スタートアドレスが指定されると共に、ステップ７２ｇで、送信元の受信ＲＡＭ１５Ｘの読取状態信号φｂｕｓｙが格納されるアドレスが設定される。受信ＲＡＭ１５Ｘに書込まれた送信元からのデータを読み出す処理を開始する際は、まず、ステップ７２ｈで、書込み状態信号φｐｕｔが格納される受信スタートアドレスのデータをチェックし、次に、ステップ７２ｉで、送信元の受信ＲＡＭ１５Ｘの読取状態信号φｂｕｓｙが格納されるアドレスにデータを送ってフラグを立てる。その後、ステップ７２ｊで転送されたデータを読取、ステップ７２ｋで、送信元の受信ＲＡＭ１５Ｘの読取状態信号φｂｕｓｙが

格納されるアドレスにデータを送ってフラグを解除する。

【００６０】

方式では双方のＶＵＰＵ１０の受信用データＲＡＭ１５Ｘに情報を持たせることが前提となるが、ＶＵＰＵ１０の間で通信が行われるので、特に制約になることではない。一方、自己の受信用データＲＡＭ１５Ｘに、相手方の状態が書き込まれているので、Ｃ言語レベルのデータを読み込む処理で相手側が読出状態、あるいは書込み状態の終了を確認できる。

【００６１】

図１９は、この方式でデータを送信するＰＵ－Ａと、データを受信するＰＵ－Ｂの間の状態情報伝達とそれを構成する信号線を示している。本例の方式では、読取状態信号φｂｕｓｙと書込状態信号φｐｕｔの専用の信号線は不要である。したがって、図１９（ｂ）に示すように、データを転送する信号線７７だけで通信チャネル３５を構成することができ、データを転送するインターフェイスのみで手順の構築が可能となる。しかしながら、その手順はプログラム側にて記載せねばならず、例えば、データ転送の回数をシーケンス番号により表示し、転送漏れが無かったかどうかをプログラム側で判断する、といった操作が必要である。

【００６２】

図２０に、本例のＶＵＰＵの変形例を示してある。このＶＵＰＵ１０Ｂは、図８に示した一般のプロセッサ３２と通信する機能を備えたＶＵ（ＣＯＭ）を備えているものである。本発明のＶＵＰＵ１０は、上述したＩＶＣ機構をＶＵＰＵ間の通信方式として採用しているものであるが、既に広く使用されているプロセッサには独自のバスプロトコルまたは通信機構を搭載している場合も多く、これら既存プロセッサとＶＵＰＵ１０を通信させることにより、さらにフレキシブルなデータ処理システム３０を構築できることは上述した通りである。すなわち、ＩＶＣ機構により複数のＶＵＰＵを用いた分散処理システムを構築したとしても、その中でひとつは既存のプロセッサを使用したいというケースも多い。このような場合にでも本発明にかかるＶＵＰＵは有効である。

【００６３】

図２０に示したＶＵＰＵ１０ＢのＶＵ（ＣＯＭ）１Ｂは、通信ユニット１２と

他のＣＰＵ３２のバスとのインターフェイスを受け持つバスブリッジ機能２６と、通信時のバッファとなるデュアルポートデータＲＡＭ２５とを備えている。また、ＶＵＰＵ１０Ｂにおいては、ＰＵ側とＶＵ側との間でレジスタ転送によるＶＵＰＵインターフェイスがサポートされているので、ＰＵ２からＶＵ１Ｂへのデータ転送はＶＵＰＵインターフェイスを利用できる。したがって、デュアルポートデータＲＡＭ２５を他のＣＰＵ３２への送信データＲＡＭとして用いることにより、ＰＵ２の側から送信を行うことができる。受信はＣＰＵ３２のシステムバスと通信ユニット１２に受信インターフェイス１４をバスブリッジすることにより受信専用データＲＡＭ１５Ｘへ書込むことができる。

【００６４】

この通信用のＶＵ（ＣＯＭ）１Ｂを設けることにより、ＶＵＰＵ１０Ｂでは、上述したＩＶＣ機能では、送信側は相手側のＶＵＰＵの受信ＲＡＭにデータを書き込むようになっているのに対し、自身の送信データＲＡＭ２５にデータを書き込むことになり、実体のある送信データＲＡＭを有するシステムとなる。したがって、ＩＶＣ機能の多くのメリットのうち、データＲＡＭの利用効率を向上できるメリットは得られない。しかしながら、既存のＣＰＵと本発明による複数のＶＵＰＵ１０による分散システム３０を構築することが可能となり、これらのタイプの異なるプロセッサが共存し、並列に各々の処理を実行可能となるメリットは大きい。

【００６５】

上記では、本発明にかかるＶＵＰＵ１０の通信ユニット１２がＰＵＴ型の場合を例に説明しているが、受信ＲＡＭ１５Ｘの代わりに送信ＲＡＭ１５Ｙを設けたＧＥＴ型であっても上記と同様のＩＶＣ機能を実現できる。図２１にＧＥＴ型の通信ユニット１２を備えたＶＵＰＵ１０をＰＵ２を中心に示してある。

【００６６】

ＧＥＴ型の場合は、ＶＵＰＵ１０に送信専用のデータＲＡＭ１５Ｙが設けられており、この送信専用のデータＲＡＭ１５Ｙが通信相手の他のＶＵＰＵ１０においては受信専用のデータＲＡＭとなる。通信ユニット１２も、送信インターフェイス１３と受信インターフェイス１４とを備えており、各々の制御部１３Ｃおよ

び１４Ｃは、送受信の条件が設定されるコンフィグレーション・レジスタ１３Ｒおよび１４Ｒを備えている。したがって、基本的な構成および動作は上記で説明したＰＵＴ型とほぼ同じである。

【００６７】

ＧＥＴ型の通信ユニット１２の調停回路１３Ａは、送信専用のデータＲＡＭ１５Ｙにデータを書き込む際に、書込み状態信号φｂｕｓｙを書込み状態にして、自分のＩＤで他のＶＵＰＵ１０に送出して書き込み状態であることを通知する。一方、送信データＲＡＭ１５Ｙからのデータの読出しは、通信先の各ＶＵＰＵ１０からのリクエスト信号あるいは読出し状態信号φｇｅｔによる。調停回路１３Ａを有する送信制御部１３Ｃは、リクエスト信号φｇｅｔが受け入れられ読出可能な状態となると、受信先のＶＵＰＵ１０のＩＤを加えた書込み状態信号φｂｕｓｙを読出し可能な状態にして送出し、通信相手のＶＵＰＵ１０へ読出可能状態であることを通知する。これにより、通信相手のＶＵＰＵ１０の受信インターフェイス１４では、アドレスを送出し、データを読み出す。したがって、ＰＵ２が通信先からデータを読み取る場合には、リクエスト信号φｇｅｔにより自分自身へのビジー信号φｂｕｓｙ（もちろん、レディ信号φｒｅａｄｙであっても良いが）を確認して、受信インターフェイス１４に示されるアドレスに応じたデータが受信制御部１４Ｃにより制御されるセレクタ１６を通じてＰＵ２に供給される。

【００６８】

送信専用のデータＲＡＭ１５Ｙも上述した受信専用のデータＲＡＭ１５Ｘと同様にデュアルポートデータＲＡＭにより構成することが可能である。この場合には送信しながら書込操作が可能となり、並列性が向上する。しかしながら、調停機能を設けない場合は、読出しと書込みのアドレスが同一の場合を考慮して、入力データＤＩを出力データＤＯにバイパスする論理回路が必要となる。

【００６９】

図２２に、ＧＥＴ型のＩＶＣ機構を実現する通信ユニット１２の動作をフローチャートで纏めてある。実際に通信を開始する前に、送信用のコンフィグレーション・レジスタ１３Ｒに、送信先のＶＵＰＵのＩＤ、送信ＲＡＭ１５Ｙの開始ア

ドレス、受信するデータの開始アドレス（実体のない受信ＲＡＭに割り当てられたアドレス）などを設定し、受信用のコンフィグレーション・レジスタ１４Ｒに、受信元となるＶＵＰＵのＩＤ、送信ＲＡＭの開始アドレス、受信されるデータの開始アドレス、などを設定する処理が行われる。これらのコンフィグレーション・レジスタ１３Ｒおよび１４Ｒの設定は、Ｃ言語のレベルであればインライン・アセンブル記述により設定できる。また、この処理をファンクションとしてサブルーチン化しておくことができる。

【００７０】

そして、プログラムにしたがって入出力アドレスが出力されると、通信ユニット１２においては、まず、ステップ８１でデータの入出力アドレスを判断する。入出力データが通常のデータＲＡＭに割り当てられたアドレスでないときは、ステップ８２で、アドレスに基づき出力処理か入力処理かを判断する。出力の場合は、ステップ８３で送信ＲＡＭ１５Ｙが読出し中でないこと、すなわち、読出し状態信号（リクエスト信号）φｇｅｔの読出し終了を待ち、ステップ８４で自己の送信ＲＡＭ１５Ｙにデータを書き込む。それと同時に書込み状態信号φｂｕｓｙを書込みにして読出しを禁止し、書込みが終了すると状態信号φｂｕｓｙを終了状態にする。

【００７１】

一方、ステップ８２で入力の場合は、リクエスト信号φｇｅｔを送出し、ステップ８５で状態信号φｂｕｓｙが書込み終了になるのを待ち、ステップ８６でデータを通信先のＶＵＰＵ１０から受領する。そして、読出しが終了すると、リクエスト信号φｇｅｔを終了状態にする。このように、ＧＥＴ型においても、入出力のアドレスによりデータを受信先のＶＵＰＵ１０のデータＲＡＭ１５Ｙから取得する制御方法を採用することにより、Ｃレベルの記述でデータの入出力アドレスを管理あるいは制御することだけで複数のＶＵＰＵ１０の間でデータを簡単に交換することができる。そして、このような処理を通信ユニット１２のファームウェアやゲートロジックで行っても良く、あるいはＣ言語のレベルで記述することも可能である。

【００７２】

上述したＰＵＴ型の通信方法とＧＥＴ型の通信方法は、どちらもＣ言語から直接データをアクセスできる点では同じであり、自己のＶＵＰＵのデータＲＡＭにアクセスするのと同じ操作で他のＶＵＰＵのデータＲＡＭにデータを書込・読出することによりデータ交換を行うことができる。ＰＵＴ型の通信方法のＶＵＰＵ１０を採用したデータ処理システム３０では、親のＶＵＰＵ１０ｐあるいは他のプロセッサが、共通のデータを複数の子のＶＵＰＵ１０ｃに転送し、子のＶＵＰＵ１０ｃは転送されたデータを頻繁にアクセスし、かつ加工し、処理を進める分散処理に適している。一方、ＧＥＴ型の通信方法のＶＵＰＵ１０を採用したデータ処理システム３０は、親のＶＵＰＵ１０ｐあるいは他のプロセッサから子のＶＵＰＵ１０ｃに供給されるデータが少量であり、さらに、子のＶＵＰＵ１０ｃがデータをそれぞれ独立に参照しながら処理を進める分散処理に適している。

【００７３】

さらに、目的に応じてＰＵＴ型とＧＥＴ型を両立させたデータ処理システムを構築することも可能である。例えば、親のＶＵＰＵ１０ｐのデータを少量ずつ複数の子のＶＵＰＵ１０ｃが参照しながら各々分散処理を行い、その結果を親のＶＵＰＵ１０ｐに戻す処理が要求される場合がある。このような処理に対応したデータ処理システム３０としては、ＧＥＴ型の通信方法で親のＶＵＰＵ１０ｐから子のＶＵＰＵ１０ｃにデータを転送し、ＰＵＴ型の通信方法で子のＶＵＰＵ１０ｃから親のＶＵＰＵ１０ｐにデータを返却する方式が最もメモリ効率がよい。なぜなら、送信専用・受信専用データＲＡＭを親のＶＵＰＵ１０ｐがひとつ持てばよいからである。また、ひとつの親のＶＵＰＵ１０ｐと複数の子のＶＵＰＵ１０ｃにより分散処理を行う構造のデータ処理システム３０は、本発明にかかるＶＵＰＵ１０を用いた極めて基本的な構造であると考えられる。したがって、親のＶＵＰＵ１０ｐにのみ転送専用メモリを保有させて共有化を図るデータ処理システムは、本発明のＶＵＰＵ１０を用いた有効な分散処理の基本構造であるといえる。

【００７４】

図２３に、送信専用のデータＲＡＭ１５Ｙと受信専用のデータＲＡＭ１５Ｘを有するＶＵＰＵ１０ｐの構成例を示してある。このＶＵＰＵ１０ｐにおいては、

通信ユニット１２の送信インターフェイス１３は上述したＧＥＴ型の構成であり、送信専用のデータＲＡＭ１５Ｙを制御し、子のＶＵＰＵ１０ｃのそれぞれからのリクエスト信号φｇｅｔに基づいてデータ転送を行う。受信インターフェイス１４は、ＰＵＴ型の構成であり、子のＶＵＰＵ１０ｃのそれぞれからの書込要求信号φｐｕｔに基づいてデータの書込を行う。

【００７５】

なお、以上では、通常のデータＲＡＭ１５Ｎ、受信専用のデータＲＡＭ１５Ｘおよび送信専用のデータＲＡＭ１５Ｙがそれぞれ独立している構成を例に説明しているが、同一のデータＲＡＭの領域を割り振ることで対応することも可能である。しかしながら、受信専用および送信専用のデータＲＡＭはデュアルポートＲＡＭあるいは多ポートＲＡＭを採用することによるメリットがあり、通信容量が小さくて良いデータ処理システムにおいては、受信専用あるいは送信専用のデータＲＡＭを独立して設けることが望ましい。

【００７６】

【発明の効果】

以上に説明したように、本発明においては、専用データ処理ユニット（ＶＵ）と汎用データ処理ユニット（ＰＵ）とを有するデータ処理装置（ＶＵＰＵ）において、ＰＵに通信機能を持たせることにより、複数のＶＵ、すなわち専用回路を並列に実行できるデータ処理システムを極めて短期間に、そして低コストで開発することができる。システムＬＳＩとして与えられた仕様全体をハードウェア化する作業は膨大であり時間と経費の点から現在ではほとんど経済的に見合わないものとなっている。これに対し、本発明のＶＵＰＵは、システムＬＳＩとして与えられた仕様の内、ハードウェア化するのに適した機能を適当な単位で抽出し、シミュレーションによって高速化などの効果が確認された機能だけをＶＵとしてハードウェア化することができる。したがって、ハードウェア化される範囲は限られたものとなり、容易に設計および開発でき、また費用も最小限で済む。その一方で、ハードウェア化したことによる効果は最大限にすることが可能である。それに加えて、ハードウェア化したＶＵを並列に実行することができるので、処理を複数のＶＵに分散することが可能となり、処理効率が高く処理速度の速い、

さらに経済的なデータ処理システムを提供することが可能となる。

【００７７】

さらに、本発明のＶＵＰＵは、繰り返し計算の多い処理などを機能単位で抽出してＶＵとして実現し高速処理を可能とすると共に、他の処理は汎用プロセッサであるＰＵで処理することによって、ハードウェア化に伴うコストアップや設計期間の長期化を抑制し、さらに、仕様変更や、開発のあらゆる段階の変更にも柔軟に対処できるというメリットも備えている。そして、プログラムレベルで制御できるＰＵに通信機能を設けることにより、プログラムレベルで並列処理の制御を行うことが可能となり、極めて柔軟な制御が可能となる。その結果、高級言語で記述された仕様に基づくシステムＬＳＩを極めて短い期間で設計および開発できる。

【００７８】

このようなＶＵＰＵを用いてデータ処理システムを構築することにより、ひとつのＣ言語などの高級言語で記述されたプロセスを分割して複数の処理プロセスにし、これらプロセス間のデータの転送と処理依頼、それに基づく処理結果の返却を設計するにはデータ転送に関してＣ言語あるいはＪＡＶＡ言語などの高級言語との親和性が高く、かつハードウェアを意識しないでデータ転送の設計が進められる方式が不可欠となる。上述した本発明によれば、アドレスによって、通信先のＶＵＰＵの受信専用のデータＲＡＭにデータを送信し、あるいは通信先のＶＵＰＵの送信専用のデータＲＡＭからデータを取得することができる。このため、ＶＵＰＵ間の通信を、メモリへのアクセスと同じ方法にてＣ言語などから直接に行うことができ、極めて自由にデータ送受を行うことができる。このため、複数のＣ言語により記述されたプロセスが並列に動作するように設計することが極めて容易となる。

【００７９】

このように、Ｃ言語あるいはその他の高級言語との連動性および対応性のあるデータ通信機構をハードウェア側に設けることにより、極めて容易に、高級言語のレベルでデータ転送が記述できる。その結果、高級言語により記述されたプロセスを複数に分割が容易となり、分散処理システムの設計が可能となる。したが

って、本発明で開示した通信機構は、上述した複数のＶＵＰＵを用いた処理速度の速いデータ処理システムを構築するのに好適なものである。

【図面の簡単な説明】

【図１】

本発明に係るＰＵおよびＶＵを備えたデータ処理装置（ＶＵＰＵ）の概要を示す図である。

【図２】

Ｃ言語により記述されたプロセスを複数に分解する様子を示す図である。

【図３】

本発明のデータ処理装置により分散処理するデータ処理システムを構築する例を示す図である。

【図４】

図３に示したデータ処理システムの各ＶＵＰＵの実行状態を例示する図である。

【図５】

Ｃ言語で記述したプログラムが分散処理用に分解される様子を示す図である。

【図６】

本発明のデータ処理装置により分散処理するデータ処理システムの異なる例を示す図である。

【図７】

本発明のデータ処理装置により分散処理するデータ処理システムのさらに異なる例を示す図である。

【図８】

本発明のデータ処理装置により分散処理するデータ処理システムのさらに異なる例を示す図である。

【図９】

Ｃ言語で記述された機能をＶＵＰＵ化する過程の概要を示す図である。

【図１０】

本発明の通信機能を有するＶＵＰＵの概略構成をＰＵの構成を中心に示す図で

ある。

【図１１】

２つのＶＵＰＵで交信する際のメモリの使用状況を示す図である。

【図１２】

親のＶＵＰＵと複数の子のＶＵＰＵで交信するデータ処理システムの概要を示す図である。

【図１３】

図１２に示すデータ処理システムの各ＰＵのメモリマップを示す図である。

【図１４】

通信ユニットの処理の概要を示すフローチャートである。

【図１５】

受信ＲＡＭにデータを入出力するタイミングを示す図である。

【図１６】

通信ユニットの処理をＣ言語により制御するプログラム例を示す図である。

【図１７】

調停を行うための状態信号とそれに対応する信号線を示す図である。

【図１８】

状態信号を受信ＲＡＭに書き込む通信方式の処理をＣ言語により制御するプログラム例を示す図である。

【図１９】

状態信号を受信ＲＡＭに書き込む通信方式における状態信号と信号線とを示す図である。

【図２０】

本発明の通信機能を有するＶＵＰＵの概略構成をＰＵの構成を中心に示す図であり、他のＣＰＵとの通信機能を備えたＶＵ（ＣＯＭ）を有するＶＵＰＵの概要を示す図である。

【図２１】

本発明の通信機能を有するＶＵＰＵの概略構成をＰＵの構成を中心に示す図であり、ＧＥＴ型の通信機能を有するＶＵＰＵを示す図である。

【図２２】

図２１に示すＶＵＰＵの通信ユニットの処理の概要を示すフローチャートである。

【図２３】

本発明の通信機能を有するＶＵＰＵの概略構成をＰＵの構成を中心に示す図であり、ＰＵＴ－ＧＥＴ型の通信機能を有するＶＵＰＵを示す図である。

【符号の説明】

１　専用データ処理ユニットＶＵ

２　汎用データ処理ユニットＰＵ

３　汎用プロセッサＰＵＸ

４　コードＲＡＭ

４ａ　制御プログラム

５　フェッチユニットＦＵ

１０　データ処理装置（ＶＵＰＵ）

１１　実行ユニット

１２　通信ユニット

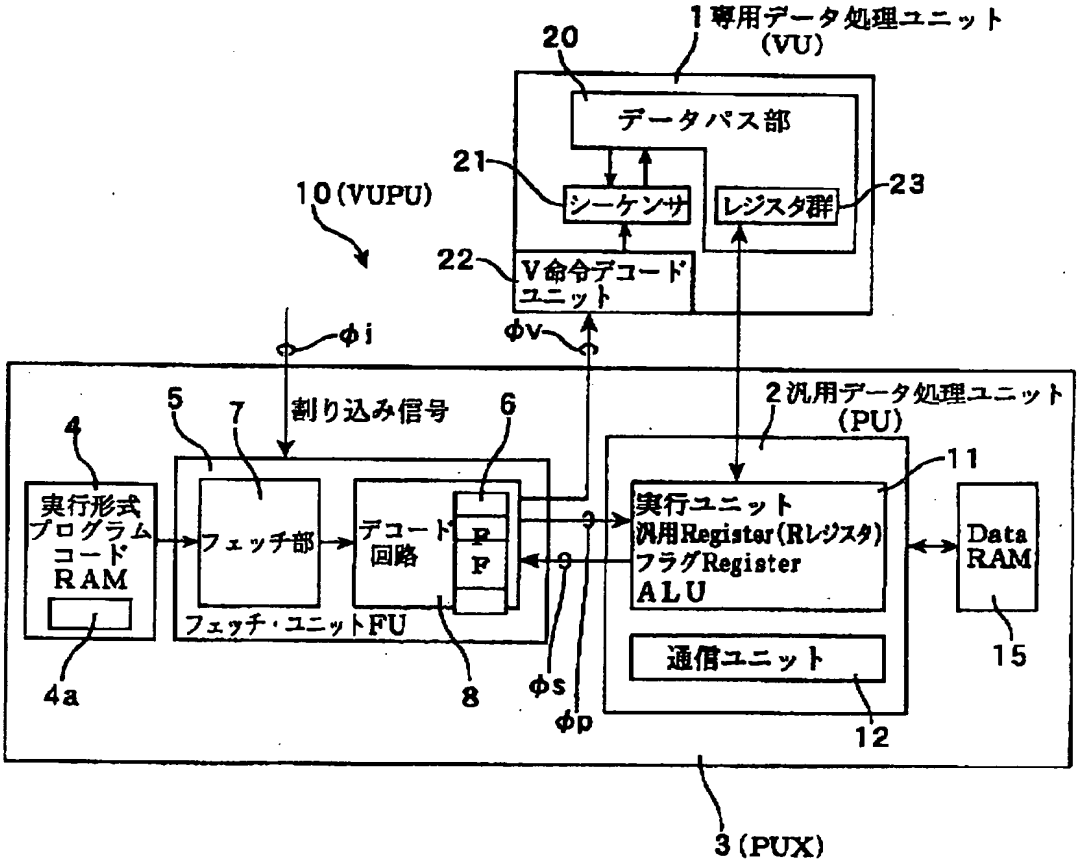
１３　送信インターフェイス、１４　受信インターフェイス

１５Ｎ　ＲＤ／ＷＲデータＲＡＭ

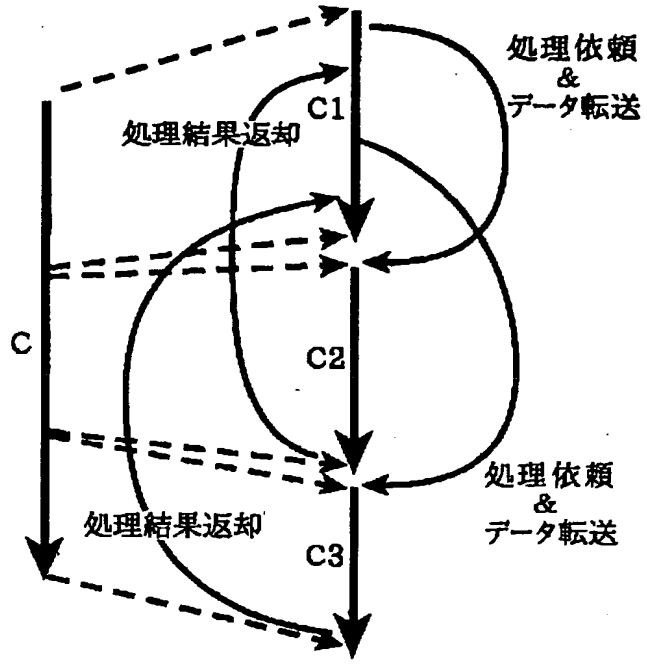
１５Ｘ　受信専用のデータＲＡＭ、１５Ｙ　送信専用のデータＲＡＭ

【書類名】　　図面

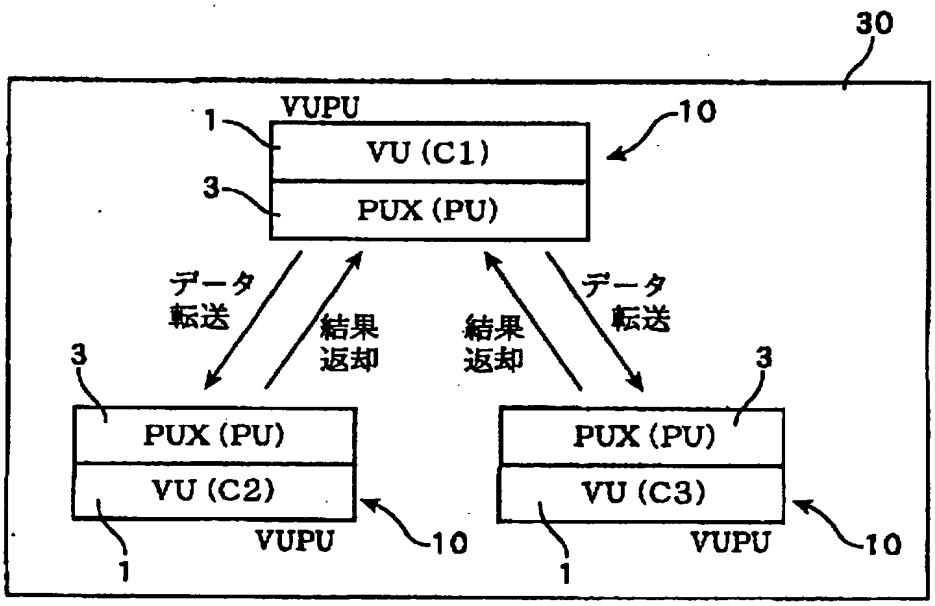
【図１】


1専用データ処理ユニット
(VU)
20
データバス部
21
シーケンサ
レジスタ群
23
10(VUPU)
22
V命令デコード
ユニット
φi
φV
割り込み信号
5
7
6
2汎用データ処理ユニット
(PU)
4
11
実行形式
プログラム
コード
RAM
フェッチ部
デコード
回路
P
P
実行ユニット
汎用Register(Rレジスタ)
フラグRegister
ALU
Data
RAM
フェッチ・ユニットFU
通信ユニット
15
4a
8
φs
φp
12
3(PUX)

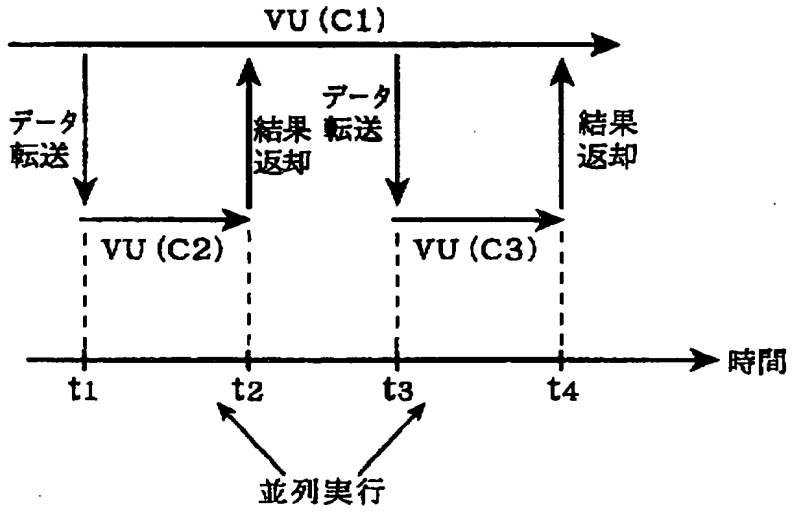
【図２】


処理依頼
&
データ転送
C1
処理結果返却
C
C2
処理依頼
&
データ転送
処理結果返却
C3

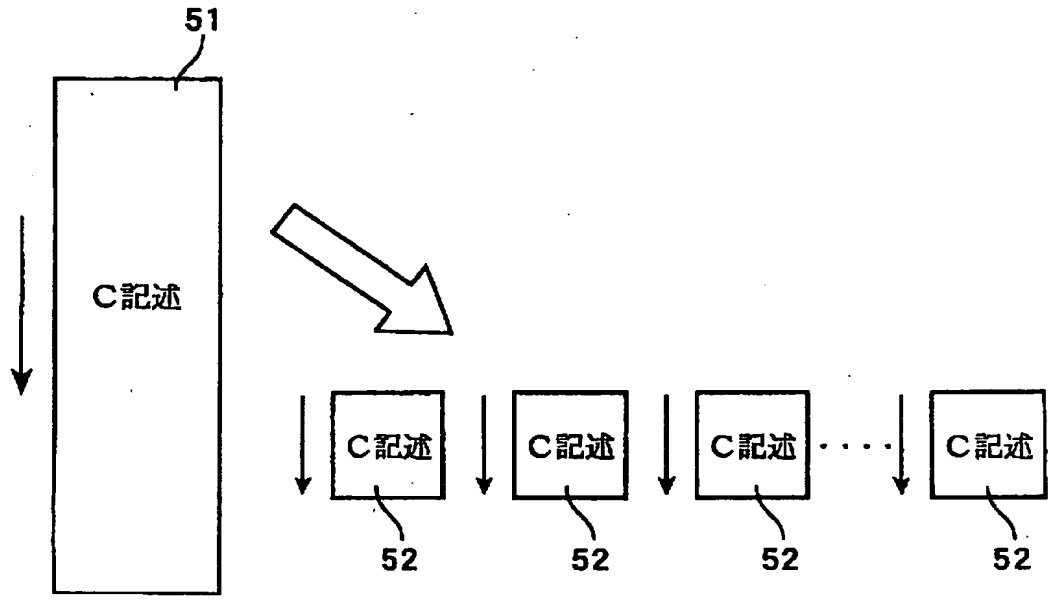
【図３】


30
VUPU
1
10
VU (C1)
3
PUX (PU)
データ
転送
結果
返却
結果
返却
データ
転送
3
3
PUX (PU)
VU (C2)
VUPU
10
1
PUX (PU)
VU (C3)
VUPU
10
1

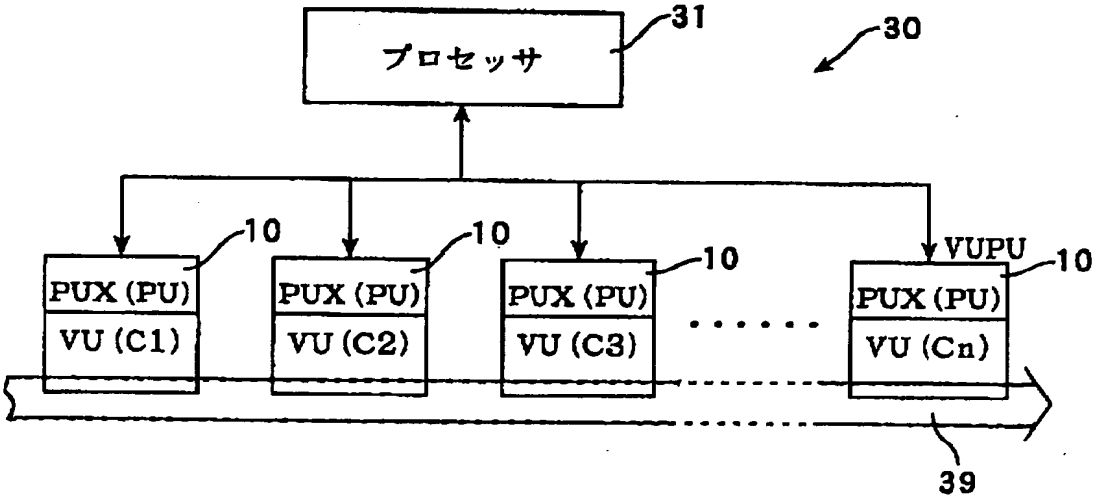
【図４】


VU (C1)
データ転送
結果返却
データ転送
結果返却
VU (C2)
VU (C3)
時間
t1
t2
t3
t4
並列実行

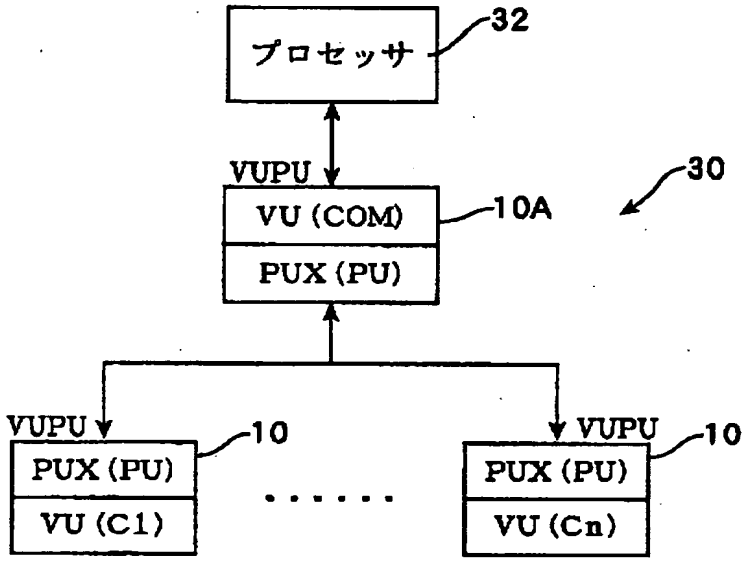
【図５】


51
C記述
C記述
C記述
C記述
C記述
52
52
52
52

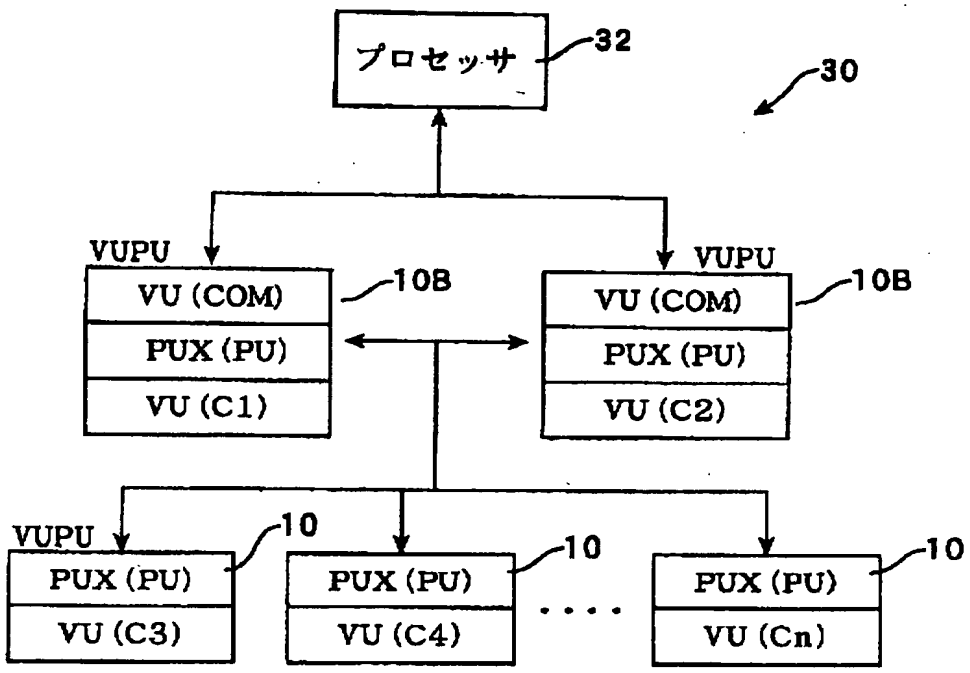
【図６】


プロセッサ
31
30
10
10
10
VUPU
10
PUX (PU)
VU (C1)
PUX (PU)
VU (C2)
PUX (PU)
VU (C3)
PUX (PU)
VU (Cn)
39

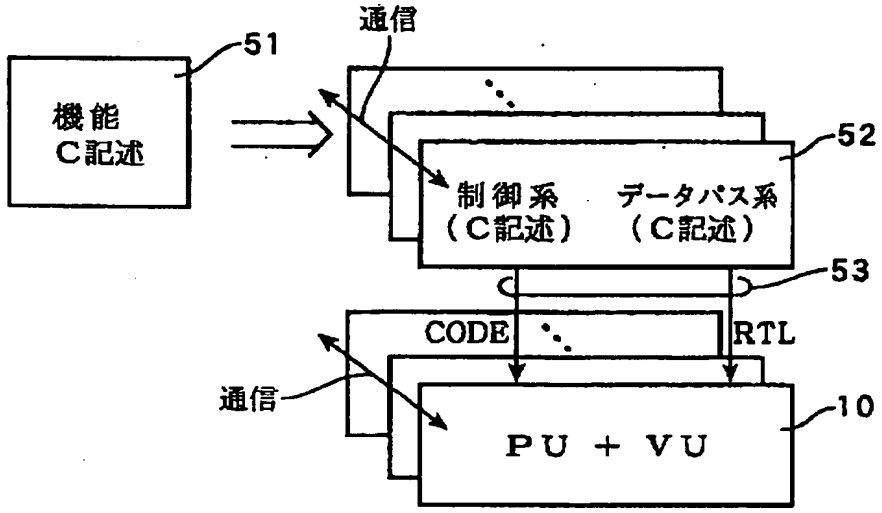
【図７】


プロセッサ
32
VUPU
30
VU (COM)
10A
PUX (PU)
VUPU
10
VUPU
10
PUX (PU)
VU (C1)
PUX (PU)
VU (Cn)

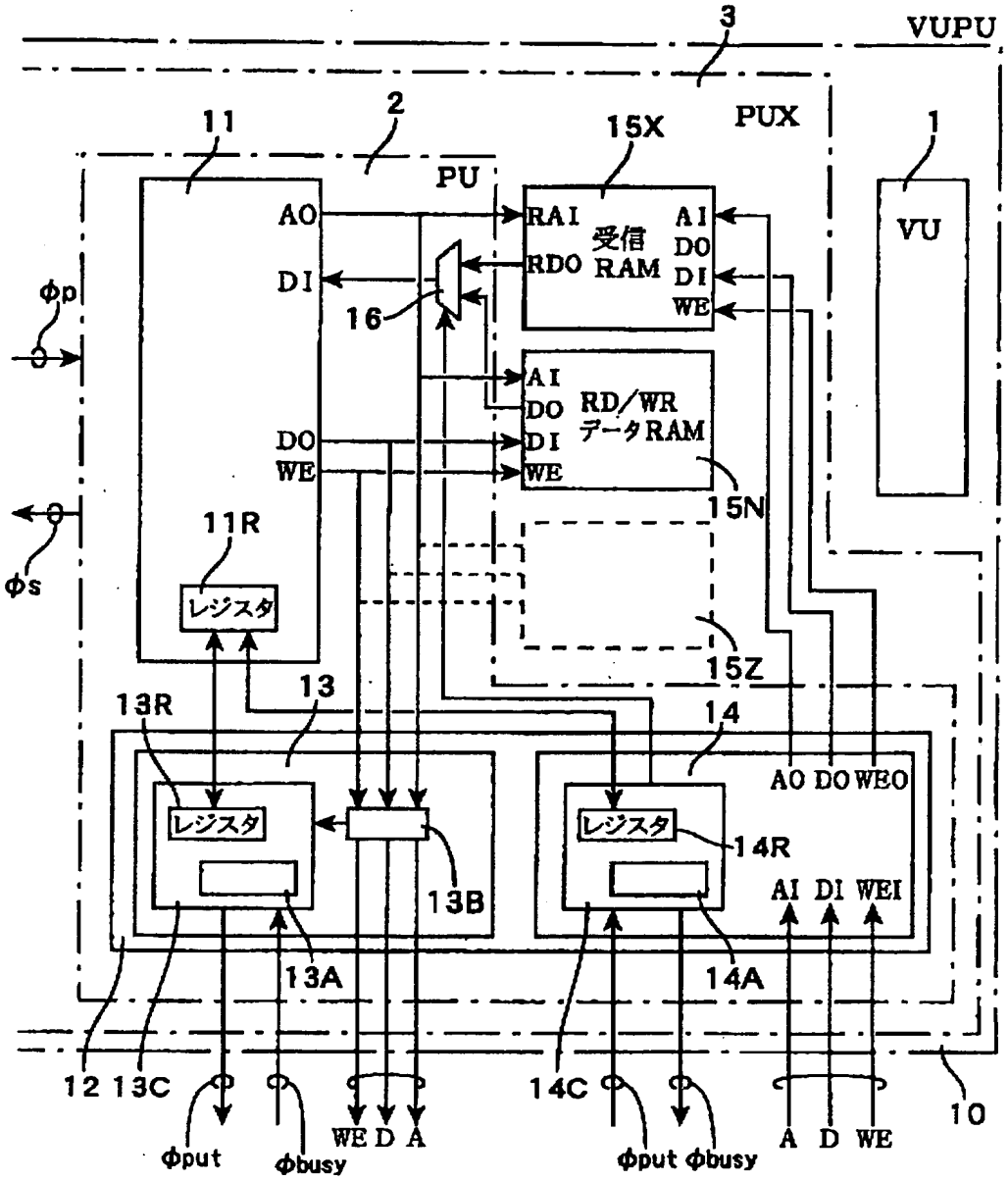
【図８】


プロセッサ
32
30
VUPU
VU (COM)
PUX (PU)
VU (C1)
10B
VUPU
VU (COM)
PUX (PU)
VU (C2)
10B
VUPU
PUX (PU)
VU (C3)
10
PUX (PU)
VU (C4)
10
PUX (PU)
VU (Cn)
10

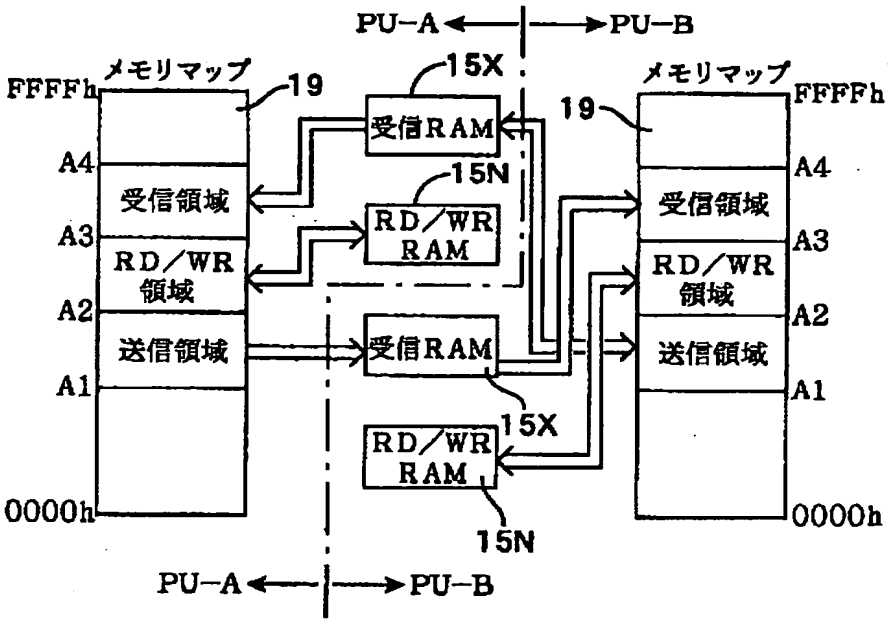
【図９】


通信
51
機能
C記述
52
制御系
（C記述）
データパス系
（C記述）
53
CODE
RTL
通信
10
PU + VU

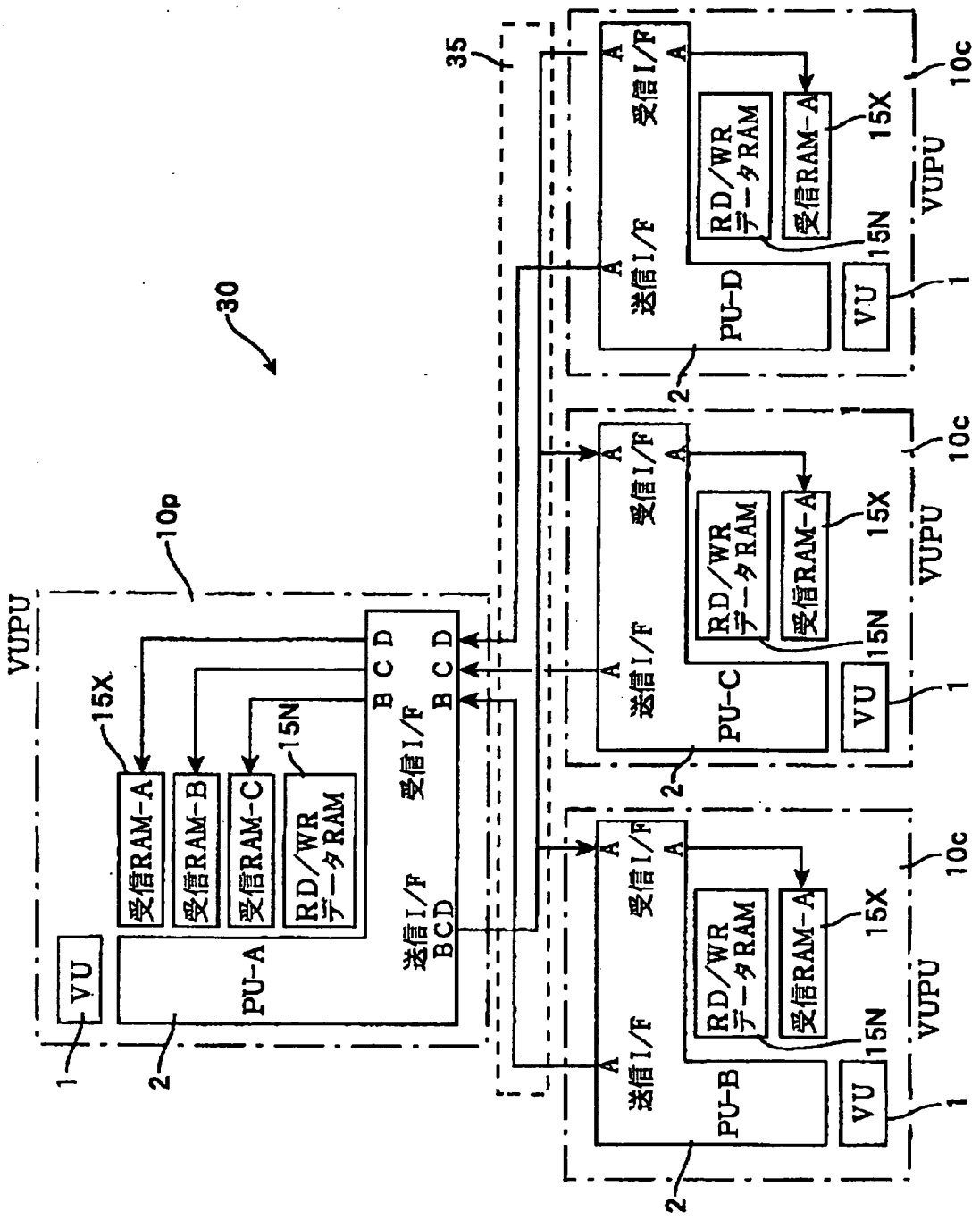
【図１０】


3
VUPU
11
2
PU
15X
PUX
1
VU
AO
DI
RAI
RDO
受信
RAM
AI
DO
DI
WE
16
φp
AI
DO
DI
WE
RD/WR
データRAM
DO
WE
15N
φs
11R
レジスタ
15Z
13R
13
14
AO DO WEO
レジスタ
レジスタ
14R
13B
AI DI WEI
13A
14A
12 13C
14C
10
WE D A
A D WE
φput
φbusy
φput φbusy

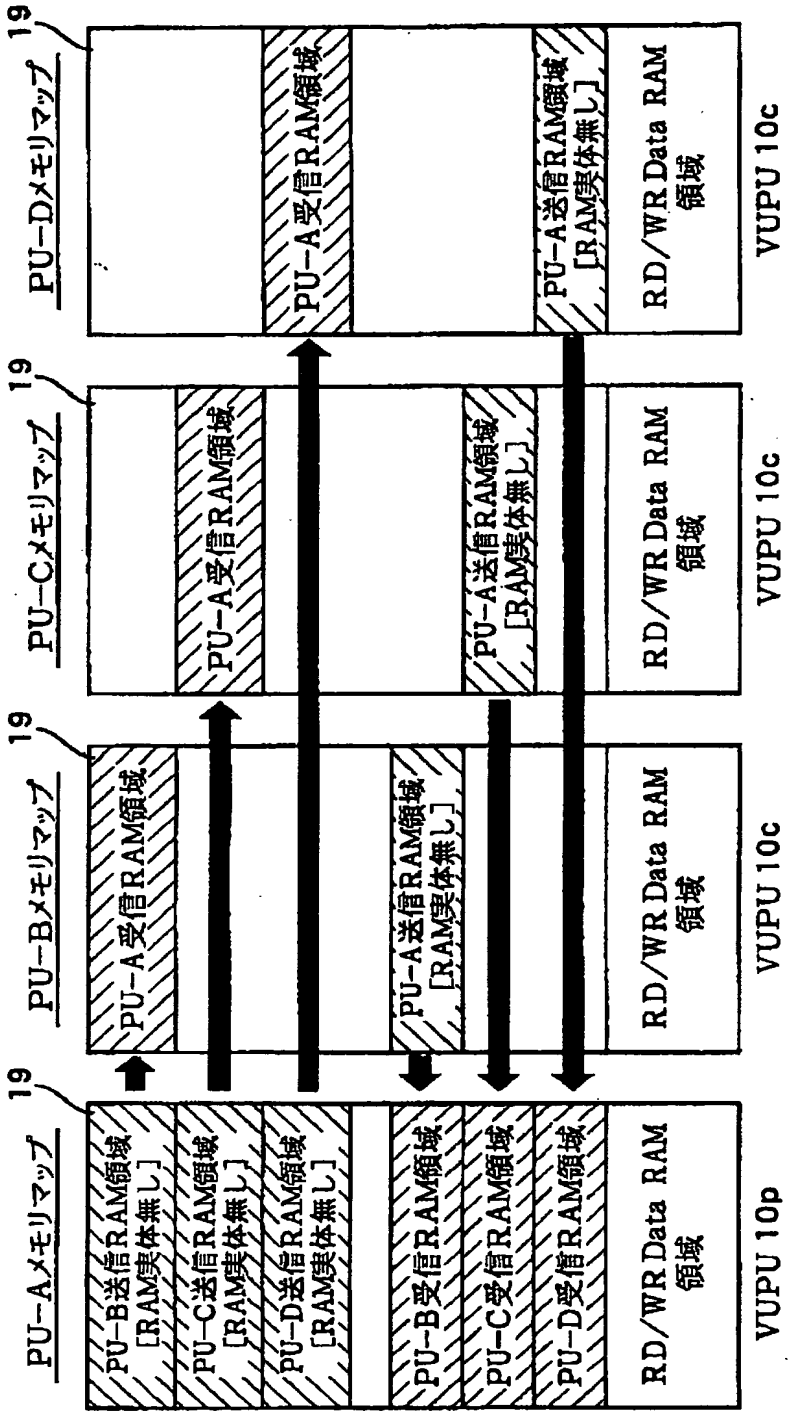
【図１１】


PU-A
PU-B
メモリマップ
FFFFh
19
15X
受信RAM
15N
RD／WR
RAM
A4
受信領域
A3
RD／WR
領域
A2
送信領域
A1
0000h
受信RAM
15X
RD／WR
RAM
15N
メモリマップ
19
FFFFh
A4
受信領域
A3
RD／WR
領域
A2
送信領域
A1
0000h
PU-A
PU-B

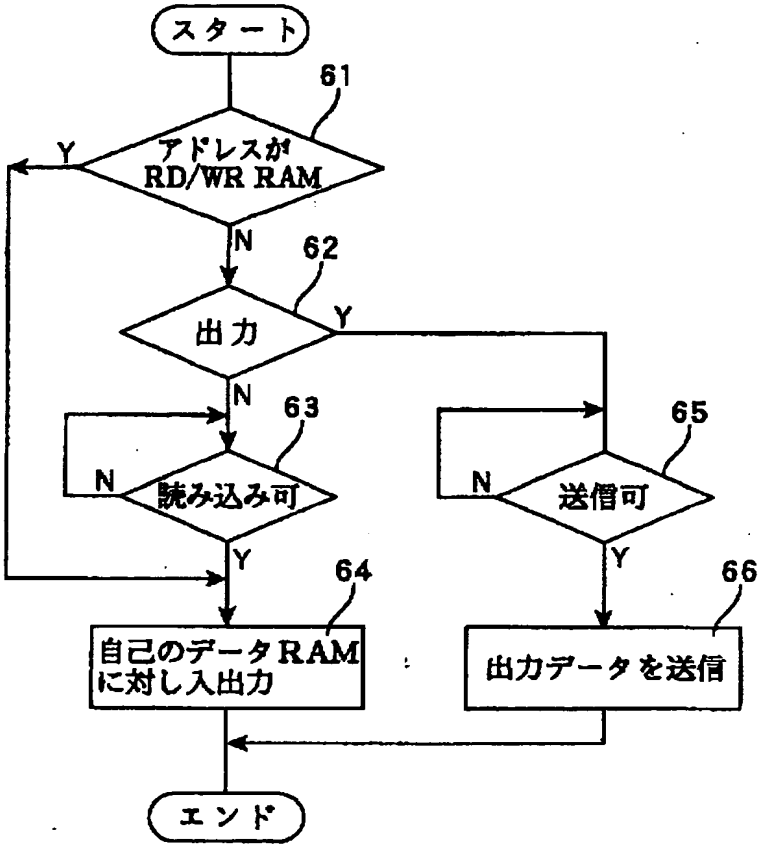
【図１２】


30
35
10p
VUPU
VU
PU-A
PU-B
PU-C
PU-D
受信RAM-A
受信RAM-B
受信RAM-C
RD／WR データRAM
15X
15N
送信I/F
受信I/F
B C D
A
10c
1
2

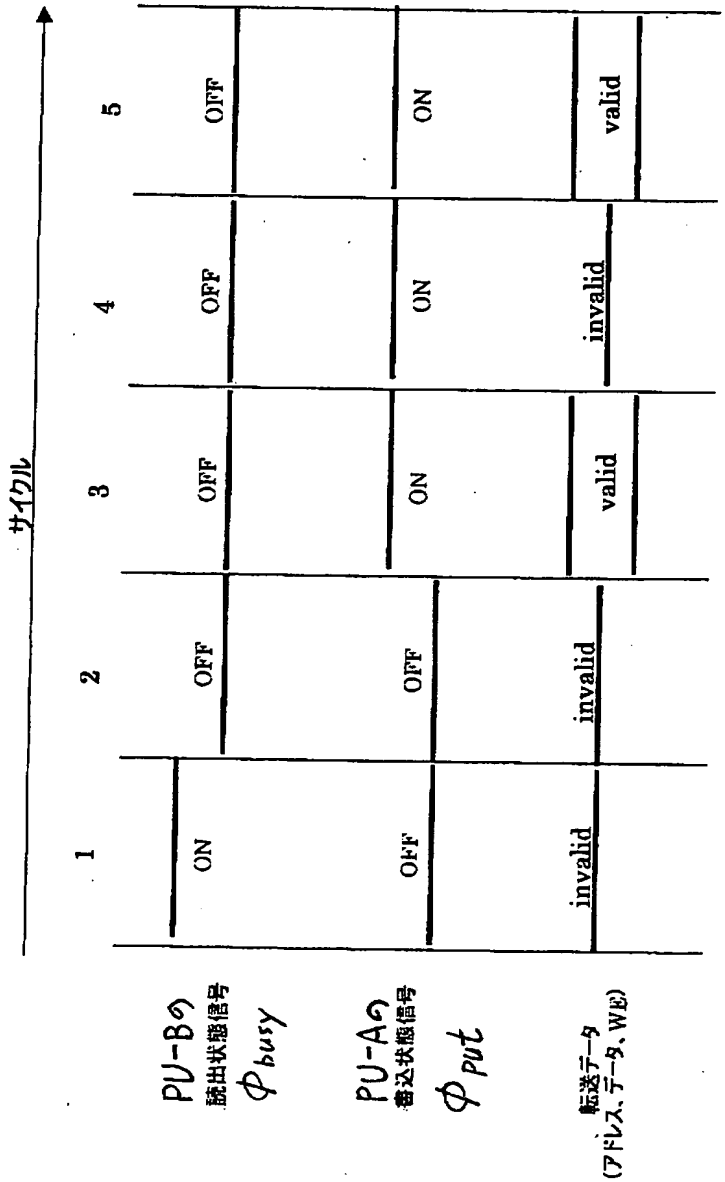
【図１３】


PU-Aメモリマップ
19
PU-B送信RAM領域 [RAM実体無し]
PU-C送信RAM領域 [RAM実体無し]
PU-D送信RAM領域 [RAM実体無し]
PU-B受信RAM領域
PU-C受信RAM領域
PU-D受信RAM領域
RD/WR Data RAM 領域
VUPU 10p
PU-Bメモリマップ
19
PU-A受信RAM領域
PU-A送信RAM領域 [RAM実体無し]
RD/WR Data RAM 領域
VUPU 10c
PU-Cメモリマップ
19
PU-A受信RAM領域
PU-A送信RAM領域 [RAM実体無し]
RD/WR Data RAM 領域
VUPU 10c
PU-Dメモリマップ
19
PU-A受信RAM領域
PU-A送信RAM領域 [RAM実体無し]
RD/WR Data RAM 領域
VUPU 10c

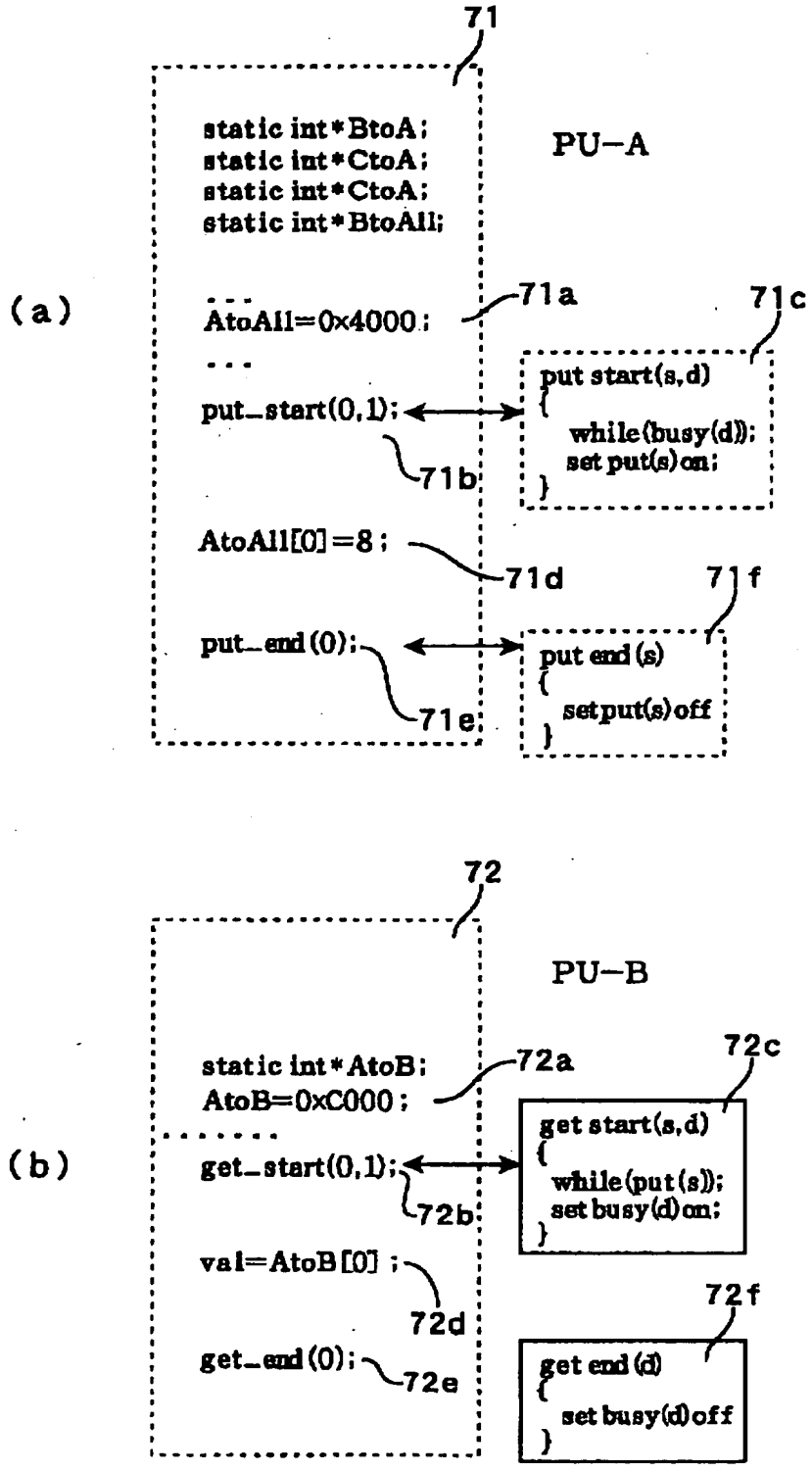
【図１４】


スタート
61
アドレスが
RD/WR RAM
Y
N
62
出力
Y
N
63
読み込み可
N
Y
64
自己のデータRAM
に対し入出力
65
送信可
N
Y
66
出力データを送信
エンド

【図１５】

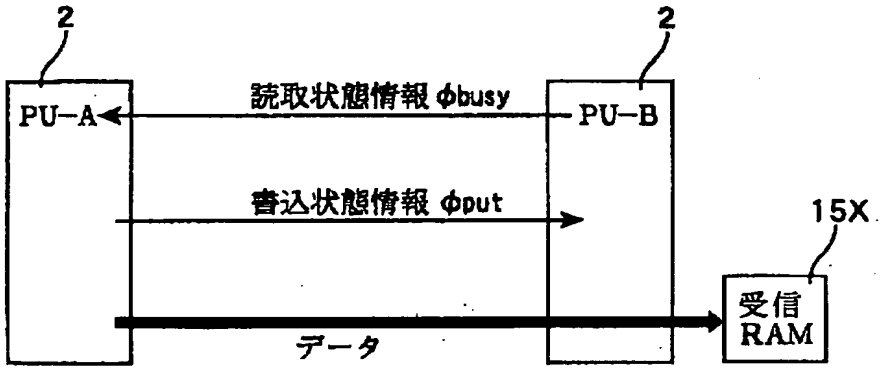

サイクル
1
2
3
4
5
PU-Bの
読出状態信号
φbusy
PU-Aの
書込状態信号
φput
転送データ
(アドレス、データ、WE)
ON
OFF
OFF
OFF
OFF
OFF
ON
ON
ON
ON
invalid
invalid
valid
invalid
valid

【図１６】

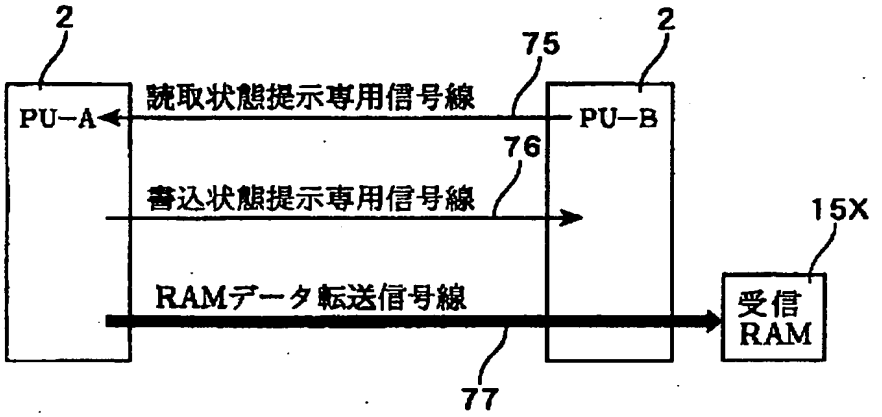

(a)
71
PU−A
static int*BtoA;
static int*CtoA;
static int*CtoA;
static int*BtoAll;
...
AtoAll=0x4000;
...
put_start(0,1);
AtoAll[0]=8;
put_end(0);
71a
71b
71c
71d
71e
71f
put start(s,d)
{
while(busy(d));
set put(s)on;
}
put end(s)
{
set put(s)off
}
(b)
72
PU−B
static int*AtoB;
AtoB=0xC000;
.....
get_start(0,1);
val=AtoB[0];
get_end(0);
72a
72b
72c
72d
72e
72f
get start(s,d)
{
while(put(s));
set busy(d)on;
}
get end(d)
{
set busy(d)off
}

【図１７】

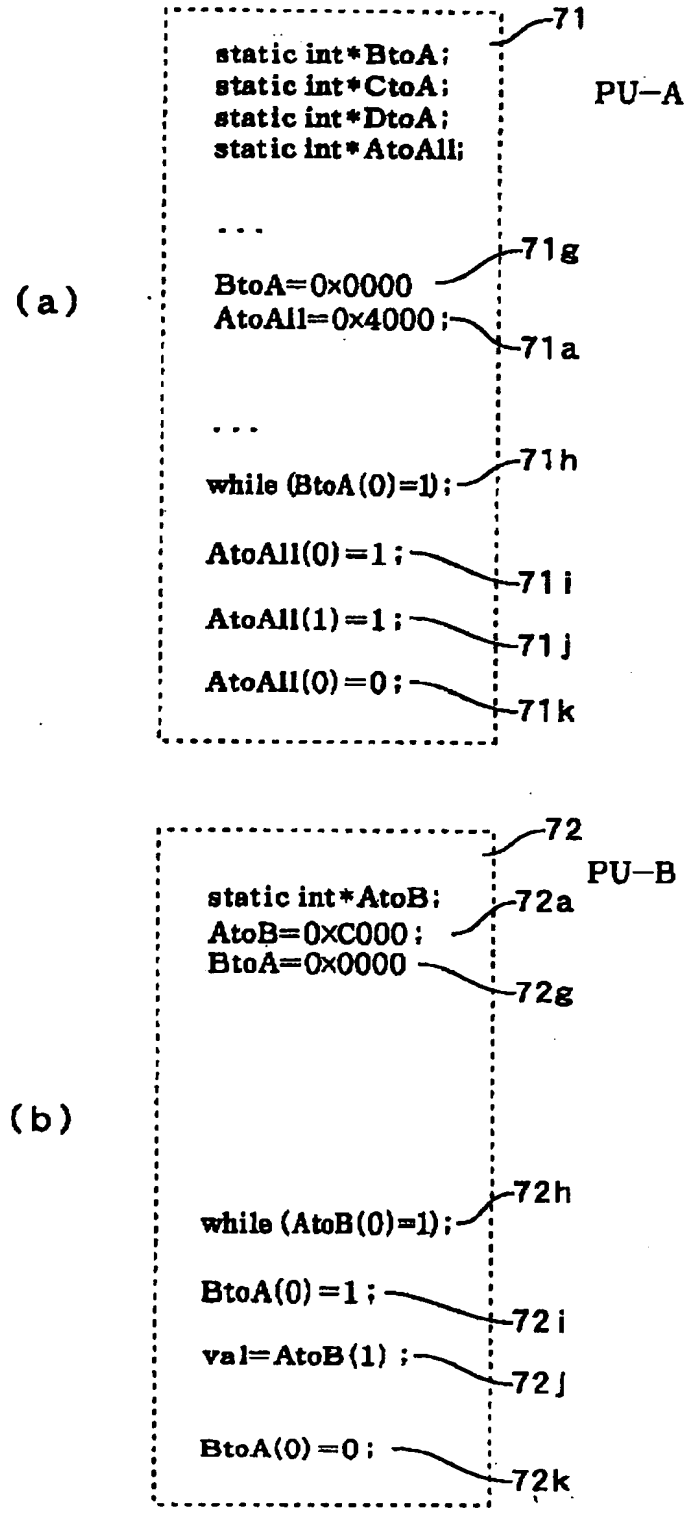
(a)


2
2
PU−A
読取状態情報 φbusy
PU−B
書込状態情報 φput
15X
受信
RAM
データ

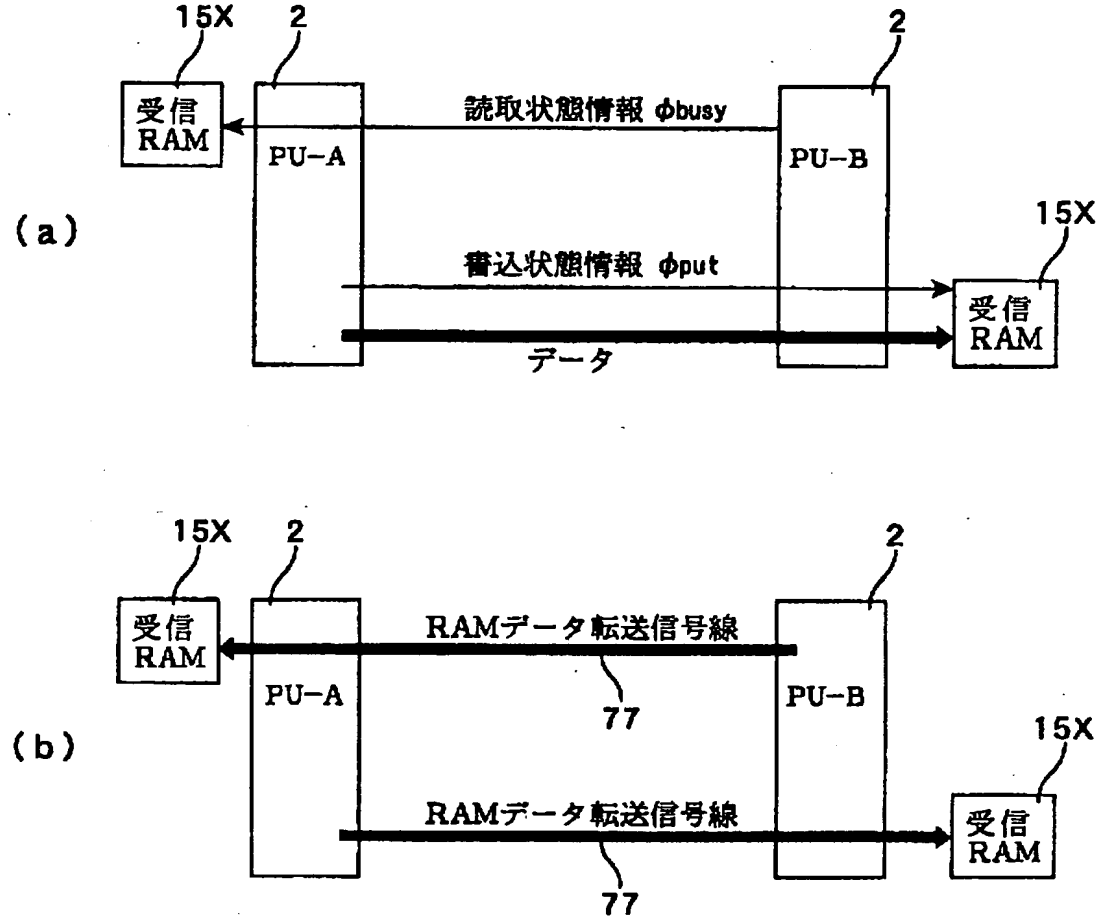
(b)


2
75
2
PU−A
読取状態提示専用信号線
PU−B
76
書込状態提示専用信号線
15X
RAMデータ転送信号線
受信
RAM
77

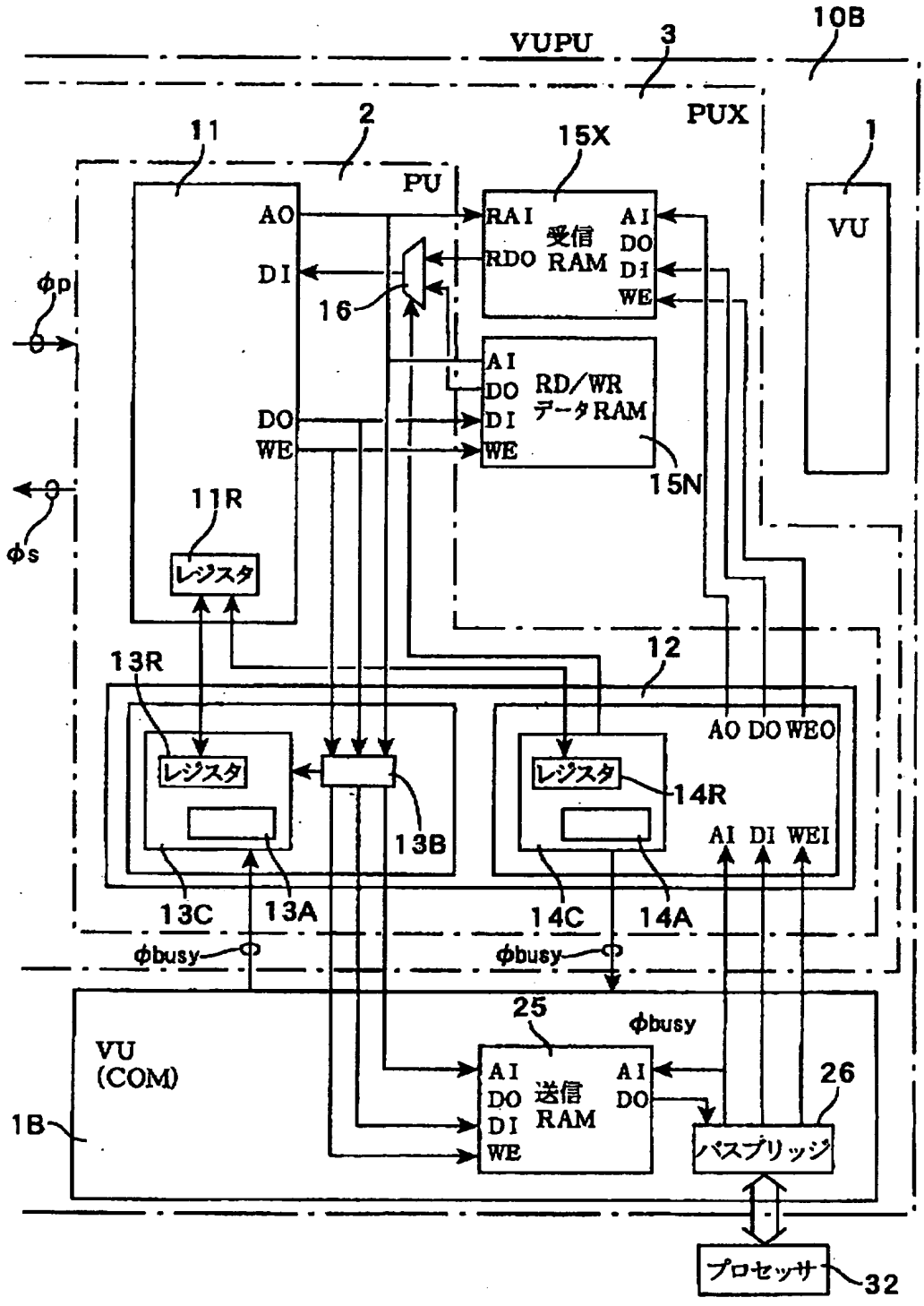
【図１８】


(a)
71
PU−A
static int*BtoA;
static int*CtoA;
static int*DtoA;
static int*AtoAll;
...
71g
BtoA=0x0000
AtoAll=0x4000;
71a
...
71h
while (BtoA(0)=1);
AtoAll(0)=1;
71i
AtoAll(1)=1;
71j
AtoAll(0)=0;
71k
(b)
72
PU−B
static int*AtoB;
72a
AtoB=0xC000;
BtoA=0x0000
72g
72h
while (AtoB(0)=1);
BtoA(0)=1;
72i
val=AtoB(1);
72j
BtoA(0)=0;
72k

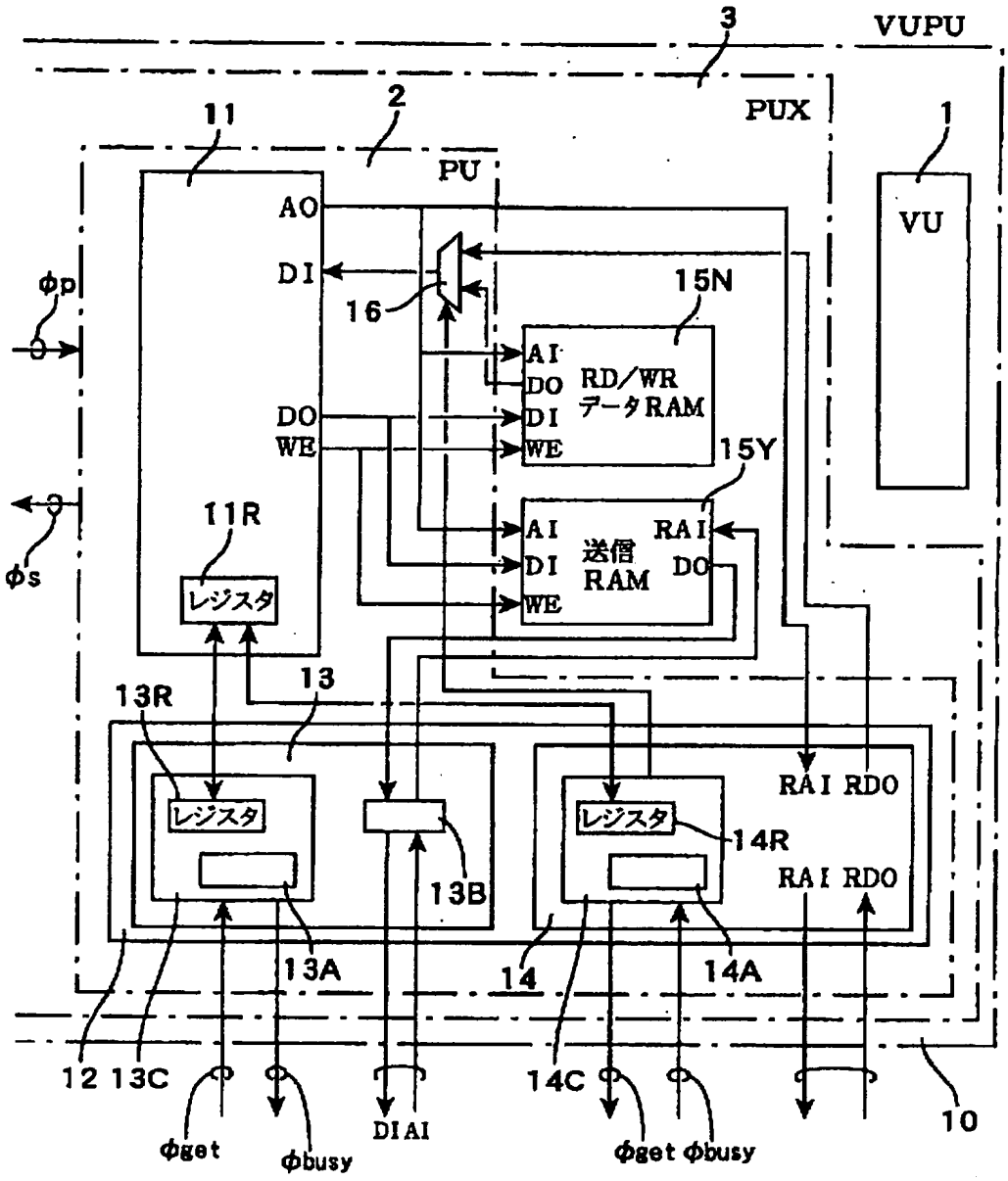
【図１９】


(a)
15X
2
受信
RAM
PU-A
読取状態情報 φbusy
2
PU-B
15X
書込状態情報 φput
受信
RAM
データ
(b)
15X
2
受信
RAM
PU-A
RAMデータ転送信号線
77
2
PU-B
15X
RAMデータ転送信号線
受信
RAM
77

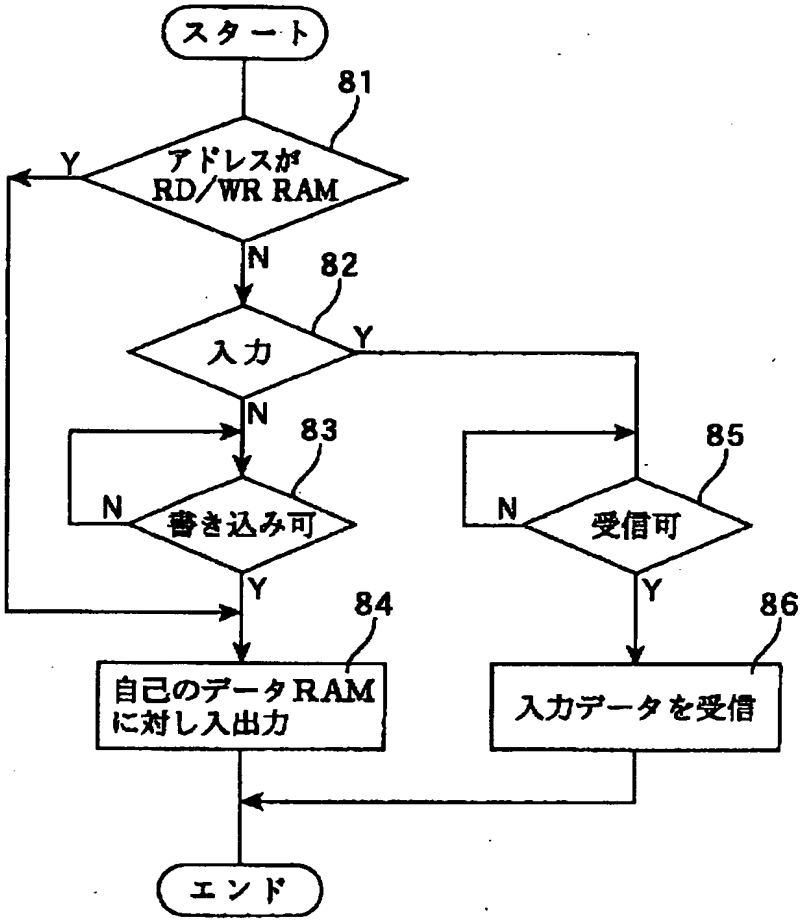
【図２０】


VUPU
3
10B
PUX
11
2
PU
15X
1
VU
AO
DI
RAI
RDO
受信
RAM
AI
DO
DI
WE
16
φp
AI
DO
DI
WE
RD/WR
データRAM
DO
WE
15N
11R
φs
レジスタ
12
13R
AO DO WEO
レジスタ
13B
レジスタ
14R
AI DI WEI
13C
13A
14C
14A
φbusy
φbusy
25
φbusy
VU
(COM)
AI
DO
DI
WE
送信
RAM
AI
DO
26
1B
バスブリッジ
プロセッサ
32

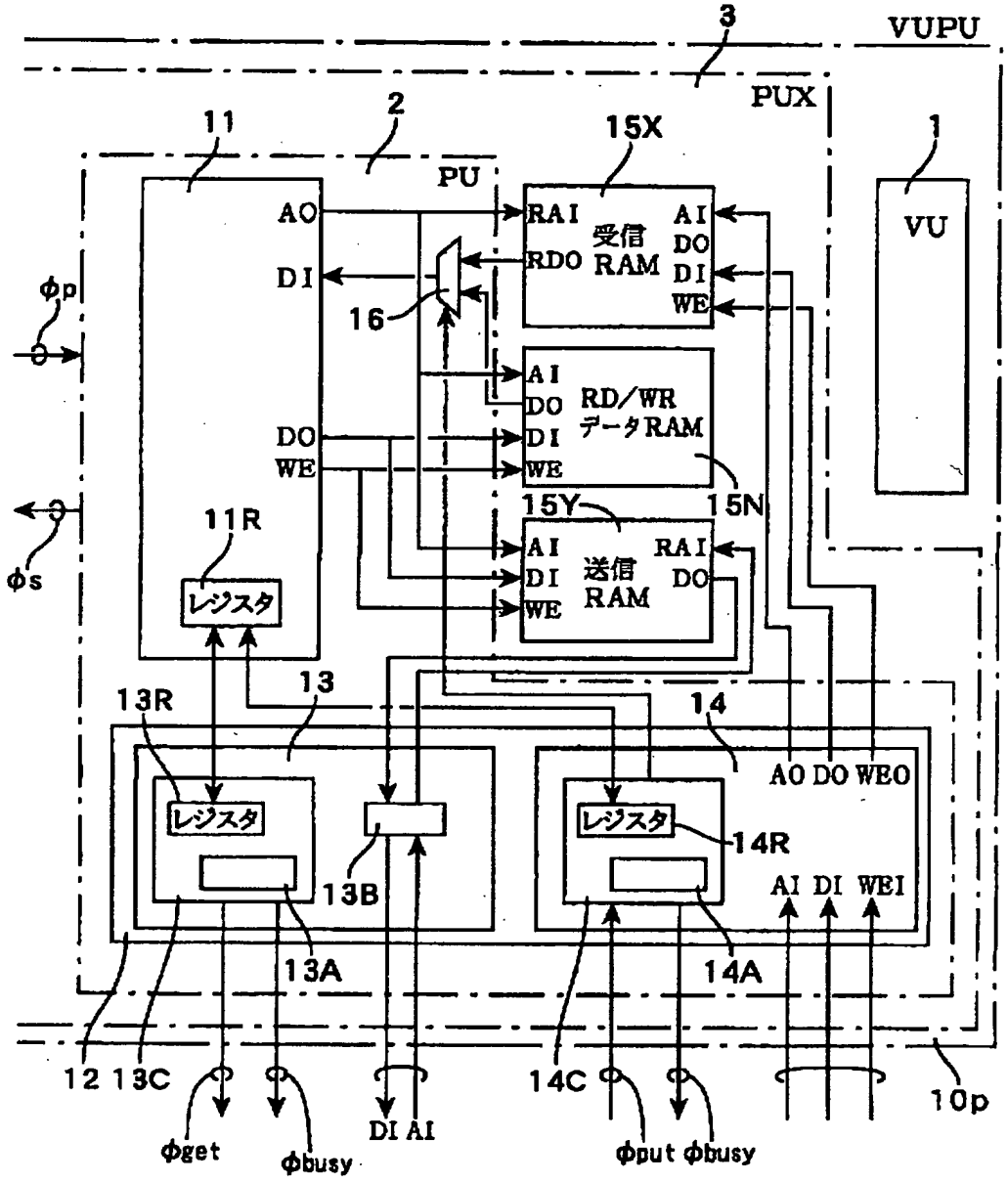
【図２１】


3
VUPU
PUX
11
2
PU
1
VU
AO
DI
16
15N
AI
DO
DI
WE
RD/WR
データRAM
DO
WE
15Y
φp
φs
11R
AI
DI
WE
送信
RAM
RAI
DO
レジスタ
13
13R
RAI RDO
レジスタ
13B
レジスタ
14R
RAI RDO
13A
14
14A
12 13C
14C
10
φget
φbusy
DI AI
φget φbusy

【図２２】


スタート
81
アドレスが
RD/WR RAM
Y
N
82
入力
Y
N
83
書き込み可
N
Y
84
自己のデータRAM
に対し入出力
85
受信可
N
Y
86
入力データを受信
エンド

【図２３】


3
VUPU
PUX
11
2
15X
1
PU
VU
AO
DI
RAI
受信
RAM
RDO
AI
DO
DI
WE
φp
16
AI
DO
DI
WE
RD/WR
データRAM
DO
WE
15Y
15N
11R
φs
AI
DI
WE
送信
RAM
RAI
DO
レジスタ
13
13R
14
AO DO WEO
レジスタ
レジスタ
14R
13B
AI DI WEI
13A
14A
12 13C
14C
10p
DI AI
φget
φbusy
φput φbusy

【書類名】　要約書

【要約】

【課題】　Ｃ言語などの高級言語で与えられる仕様を高速で処理可能なデータ処理システムを短期間で経済的に構築可能とする。

【解決手段】　汎用処理が可能な汎用データ処理ユニットＰＵ２と、特定のデータ処理をハードウェア化して高速で実行可能な専用データ処理ユニットＶＵ１とを有するデータ処理装置１０において、汎用データ処理ユニットＰＵ２に通信ユニット１２を設けることにより、複数の専用データ処理ユニットＶＵ１が並列に稼動できるデータ処理システムを構築可能とする。

【選択図】　図１０

認定・付加情報

特許出願の番号	特願２００１－０２４５１３
受付番号	５０１００１３７７１４
書類名	特許願
担当官	第七担当上席　００９６
作成日	平成１３年　２月　１日

＜認定情報・付加情報＞

【提出日】　平成13年 1月31日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔５９８１４９２４２〕

１．変更年月日　　１９９８年１０月２９日
［変更理由］　新規登録
住　所　東京都新宿区西新宿６丁目１２番１号
氏　名　パシフィック・デザイン株式会社